TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

JOONAS VIRKKI

PREDICTING THE EFFECTS OF SYSTEM CHANGES ON PASSIVE RADAR COVERAGE

Master of Science thesis

Examiners: Lecturer Simo Ali-Löytty,
Asst Prof. Lassi Paunonen
Examiner and topic approved by the
Faculty Council of the Faculty of
Natural Sciences
on 7th September 2016

ABSTRACT

JOONAS VIRKKI: Predicting the Effects of System Changes on Passive Radar Coverage
Tampere University of Technology
Master of Science thesis, 53 pages, 34 Appendix pages
February 2017
Master's Degree Programme in Science and Engineering
Major: Mathematics
Examiners: Lecturer Simo Ali-Löytty, Asst Prof. Lassi Paunonen
Keywords: Passive radar, passive coherent location, coverage, Cramér-Rao lower bound, Fisher information, modularization

Passive coherent location, or passive radar, in most cases an undetectable system to identify and track physical objects has in general a coverage much more complex than that of a tradition radar. Predicting the coverage and therefore the areas lacking vision is time-consuming to compute with reasonable accuracy, and requires complete recalculations if the system topology changes.

In this work a generalised Cramér-Rao bound for joint position and velocity estimations is used to compute visual presentations of unbiased estimator parameter variance lower bound. The estimations are combined in differing ways to qualitatively mimic modularised coverage in case of topological change in the system. The combinations and full single-simulation results are compared by the area limited by a chosen lower bound of the variance.

The areal differences in single computations versus partitionedly computed predictions range from subtle to significant for a specific limit, and especially the partly recomputed predictions have unmistakably poor results with some configurations. In fact, the type of change and system configuration seem to have the most effect on the accuracy, since some predictions achieve less that 10% difference regardless of the bound used. The predictions are smooth to a degree that substantial interpolation of data point yields only minuscule difference. This is accredited to the simplified model of signal propagation used.

The thesis demonstrates the non-linearity of coverage predictions. Primarily the methods evaluated did not achieve desirable results, and the outcome could be described as a suggestive outline of the way to approach the problem in further research. Not much could be said about the feasibility of similar modularisation without validated data, and more in-depth analysis of the lower bound.

TIIVISTELMÄ

JOONAS VIRKKI: Passiivisen tutkan järjestelmämuutosten vaikutukset toiminta-alueeseen
Tampereen teknillinen yliopisto
Diplomityö, 53 sivua, 34 liitesivua
Helmikuu 2017
Teknis-luonnontieteellinen tutkinto-ohjelma
Pääaine: Matematiikka
Tarkastajat: Yliop.leht. Simo Ali-Löytty, Assistant Professor Lassi Paunonen
Avainsanat: passiivinen tutka, peittoalue, Cramér-Rao -alaraja, Fisher-informaatio, modularisointi

Passiivinen tutka on huomaamaton kiinteiden kohteiden havaitsemiseen ja jäljittämiseen käytetty järjestelmä. Sen mallinnettu peittoalue poikkeaa monimutkaisuudessaan perinteisen tutkan kantamasta. Peittoalueen tarkka ennustaminen ja katvealueiden havaitseminen ovat laskennallisesti vaativia tehtäviä, ja koko alue on mallinnettava uudestaan, jos esimerkiksi antennien sijainnissa tapahtuu muutoksia.

Tässä diplomityössä kuvataan harhattoman estimaattorin ennusteiden tarkkuutta käyttäen yleistettyä Cramér-Rao alarajaa kohteen nopeuden ja sijainnin laskentaan. Peittoalueen jakamista erillään laskettaviksi osiksi jäljitellään erilaisilla laskentadatan yhdistelmillä tapauksissa, joissa järjestelmän lähettimien lukumäärä muuttuu. Yhdistelmien ja yhtenäisten simulaatioiden rajaamia peittoalueita vertaillaan määritellyillä tarkkuuden ylärajoilla.

Alueelliset erot yhtenäisten ja yhdistettyjen ennusteiden välillä vaihtelevat juuri havaittavasta merkittävään. Muutoksen tyyppi ja järjestelmän konfiguraatio vaikuttavat olevan keskeisiä tarkkuuden muuttujia: toiset ennusteet saavuttavat alle 10% suhteellisen eron pinta-alassa riippumatta käytetystä ylärajasta. Tarkkuuden alarajoissa ei ilmene äkillisiä poikkeamia, jolloin datan huomattavasta interpoloinnista seuraa vain häviävän pieniä eroja peittoalueeseen. Aiempien tutkimusten pohjalta ennusteiden sileyden voidaan olettaa johtuvan käytetystä, yksinkertaistetusta signaalin etenemismallista.

Työn tulokset havainnollistavat peittoalueiden ennusteiden epälineaarisen luonteen. Pääosin tutkitut metodit eivät tuota toivottuja ennusteita, ja tuloksia voi kuvailla ensisijaisesti suuntaa-antavina aiheen jatkotutkimukselle. Työn tulokset ovat linjassa keskenään, mutta ilman Cramér-Rao alarajan analyyttistä tutkimusta ja mittaamalla varmennettuja tuloksia on vaikea kuvailla modularisoinnin mahdollisuuksia.

PREFACE

This thesis was written for Insta DefSec Oy during the fall of 2016 and the following winter. I was not familiar with the topic earlier, and it offered great opportunities to learn and to keep discovering. In the end I feel the surface was barely scratched.

Most of the mathematics in the thesis are introductory and can be understood with the basics of differential and matrix calculus, in addition to the fundamentals of probability theory. The intended audience is the people with general interest in passive radars and the mathematical complexity of accuracy estimations. More advanced readers might be interested in the results and their relevance to other publications.

The examiners and my instructors at the Tampere University of Technology: university lecturer Simo Ali-Löytty and assistant professor Lassi Paunonen deserve recognition and my deepest gratitude for their efforts, insight and guidance during the process. Without their help much would have been left unlearned and I would still be thinking in circles. Thank you.

For making the work on this thesis possible I thank my colleagues at Insta DefSec including, but not limited to, Janne Ropponen, Yrjö Kinnunen and especially Antti Asiala, who provided the general topic and framework for the thesis.

My years at the Tampere University of Technology culminating in finishing this thesis have been more fruitful and rewarding than I could have ever imagined. I am immensely indebted to the good people and fellow board members of the Guild of Science and Engineering Hiukkanen, my dear friends on IRC, and all of the people, both student and personnel, that are too numerous to be named individually, with whom I have had the pleasure of spending these past years, and hopefully will have for years to come. Without you I would not be a person finishing his thesis.

Last but certainly not least, I would like to express my indisputable gratitude towards my family: my mother, father and sister who were never short of encouraging, significant words when the process felt overwhelming.

"*Mies voi lähteä Herwannasta, mutta Herwanta ei miehestä.*" -old Herwoodian proverb.

Tampere, January 19th of 2017

Joonas "Yawnas" Virkki

CONTENTS

LIST OF ABBREVIATIONS AND SYMBOLS

AREPS	Advanced Refractive Effects Prediction System
CRLB	Cramér-Rao lower bound
CSC	CSC - IT Center for Science Ltd.
FIM	Fisher information matrix
ITM	Irregular Terrain Model
MSE	Mean squared error
PDF	Probability density function
RMSE	Root-mean-squared error
SNR	Signal-to-noise ratio
SCNR	Signal-to-clutter-plus-noise ratio
VHF	Very High Frequency

$\wedge$	Logical "And"
$\mid$	"Given", logical prior assumption
$\otimes$	Kronecker product
$\nabla_{\boldsymbol{a}}$	Gradient in respect to vector $\boldsymbol{a}$.
$\triangleq$	Equal by definition
$\sim$	Has a probability distribution
$\succeq$	Succeeds in positive semi-definite partial ordering
$\langle\bullet\rangle_{a,b}$	Expected value of a random variable, while a, b are constant
$(\bullet)^H$	Conjugate transpose
$(\bullet)^T$	Transpose
$\| \bullet \|$	Euclidean distance
$B(\bullet)$	Bias
$\mathrm{cov}(\bullet, \bullet)$	Covariance of a random variables
$\mathcal{CN}(\mu, R)$	Complex Gaussian distribution with mean μ and covariance R
D	Distance between illuminator and receiver
d_i	Distance between illuminator and target
d_r	Distance between receiver and target
$\mathrm{Det}(\bullet)$	Determinant
$\mathrm{Diag}(\bullet, \ldots, \bullet)$	Block diagonal matrix with matrices in parenthesis on the diagonal
F	Receiver effective noise figure
f	Doppler shift
G_r	Receiver antenna gain
G_t	Transmit antenna gain
$\boldsymbol{I}$	Fisher information

k_B	Boltzmann constant
$\mathcal{L}(\bullet)$	Likelihood function
$L(\bullet)$	Log-likelihood function
L_s	System loss
λ	Wavelength
$\mathcal{N}(\mu, R)$	Gaussian distribution with mean μ and variance R
Ω	Sample space
$P(\bullet)$	Cumulative distribution function
$p(\bullet)$	Probability density function
$\mathbb{P}(\bullet)$	Probability measure
P_n	Receiver noise power
P_r	Received signal power
P_s	Transmit power
$S(\bullet)$	Score function of a random variable
$s(\bullet)$	Time sampled wave form
σ	Target cross-section
$\boldsymbol{\sigma}^2$	Variance
T_0	Noise reference temperature, 290 K unless stated otherwise
$\mathrm{Tr}(\bullet)$	Trace of a matrix
τ	Time delay
$\mathrm{var}(\bullet)$	Variance of a random variable
$\mathrm{vec}(\bullet)$	Column vectorizing

1. INTRODUCTION

For aerial surveillance one of the most useful concepts since its first implementations has been the radio detection and ranging, or radar. For any data gathering in general, in addition to deduced information, it is of utmost importance to avoid false extrapolation, and to know what information cannot be derived from the data. Predicting the area of operation of a distributed radar system is a complex and computationally intensive task. With distributed system changes may occur, that require the complete area to be recalculated. The aim of this thesis is to compare and evaluate different methods to expedite the prediction process after a change. For instance, could the areal effects of shutting one transmitter down be limited.

The radar works on the principle of electromagnetic radiation of some wavelength reflecting from solid surfaces but passing through gases, like atmosphere. These echoes may then be used to calculate various spatial attributes about the reflecting object. This functionality can be simplified and expressed with the radar equation. [23] Unfortunately, the world around the radar is not usually homogeneous even when monitoring objects in the atmosphere, and the electromagnetic waves propagate in a multitude of ways, making the performance of the radar a function of the prevailing environment. It is therefore a complicated task to predict the radars detection range in great accuracy, and this leads to uncertainty in what can and cannot be deduced from the radar data. Yet the range needs to be estimated in some way to achieve an effective radar coverage, since not knowing the operative area of the radar could lead to unwanted dead angles in the monitoring system.

Passive coherent location as well as passive, bistatic or multistatic radar are common names describing radar systems, where the signal transmitters are geographically separated from the receivers. The system configurations may vary from bistatic systems, which has one receiver and one transmitter to multistatic systems having an arbitrary amount of receivers and transmitters. It is not uncommon to have the transmitters operated by an oblivious third-party, such as FM radio broadcasters or telecommunications service providers [25], and to utilize electromagnetic radiation transmitted whether or not a radar system is on-line, hence “passively” observing wanted objects. The signal content, or data transmitted are not primarily of interest,

and the transmitters are used solely to illuminate the wanted targets. Therefore in context of passive radars the transmitters are usually referred to as *illuminators of opportunity* [2, 18].

Observing moving or stationary targets by comparing received and reflected signals from possibly multiple illuminators utilizes advanced and computationally intensive signal processing and statistical algorithms. Since the theory behind observing target with separate signal transmitter and receiver predates combined radar technology, the theory is well known. However, only recently has the rapid increase in computational power rendered it technologically advantageous. Many commercial and academic systems are already operated and researched around the globe, with results being published on target detection and tracking [13, 21], topology optimization [2, 4] and fidelity improvement [1].

The limiting factor in detecting an object is the *signal-to-noise ratio* (SNR) of the signal reflected by the target. The radar system detects objects that reflect a given illuminating signal by identifying the source of the reflected signal, in traditional monostatic radars the radar itself. Since the location of the illuminator is assumed to be known, delay, Doppler-shift and depending on the antenna configuration, the direction of the reflected signal can be used to estimate the position and velocity of the reflecting surface.

To optimize the performance of multistatic radar system the coverage given by the configuration of illuminators and receivers has to be simulated by combining the statistical algorithms with various scale of signal propagation models, such as *Advanced Refractive Effects Prediction System* (AREPS) [19], *Irregular Terrain Model* (ITM) [12] or, when modelling purely theoretical situations, bare free-space path loss, where the properties of the region are isotropic, homogeneous and loss-free [23, p.26.4]. Depending on the complexity of the chosen propagation model, calculating coverage area or three dimensional coverage space can be time-consuming, like one of the coverage prediction presented in [1], which took over an hour. Thus a dynamic model for calculating the coverage in case of changes in system topology, such as an illuminator shutting down, would be a valuable asset with real-time aerial surveillance systems. The difference between a single simulation and a simple combination of separate simulations is visualised in Figure 1.1. On the right the system of three illuminators is computed as one, such that every computed point includes the wanted signals and noise of all illuminators. On the left the corresponding estimated parameters of three systems of one illuminator are combined. The predictions are notably different. For instance the areas with higher variance in blue and green extend much closer to the receiver on the opposite side of every illuminator on the

left, whereas the prediction on the right has no such distinct “spikes”. This is one possible method to combine the predictions, and the goal is to find a way make combinations similar to the prediction on the left.

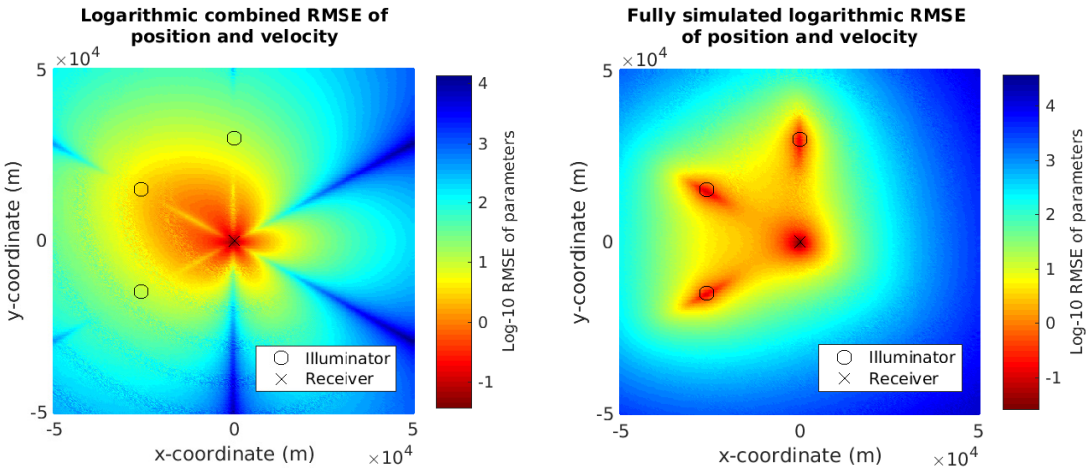

Figure 1.1 *On the left: Combined RMSE of three single pairs. On the right: Same topology simulated as a whole*

In this thesis the effects of topological changes in the system on the computational coverage of target position and velocity estimations is evaluated using a free-space propagation model. This includes analysis whether splitting the coverage calculations separately for each receiver-illuminator pair yields statistically valid results in the high SNR region, and how the recalculated area may be limited. To simplify, how to generate the right hand side on Figure 1.1 from the left hand side. The theory begins with physical maximum coverage: the simple range of the ratio between the wanted decaying signal, and the unwanted signals from the bistatic radar equation. It then continues with Bayesian statistics and Fisher information moving on to more complex statistical methods concluding in a sligthly modified *Cramér-Rao lower bound* (CRLB) estimation algorithm for the position and velocity of a target based on the algorithm by He *et al.* [10]. CRLB bounds the variance of estimated parameters, and can thus be used to estimate the maximum resolution with given parameters. Although other similar bounds exist by McAulay and Hofstetter 1971, Kendall and Stuart 1979, Seidman 1970 and Ziv and Zakai 1969, [14, p.27] Cramér-Rao bound was chosen for existing research, and for allowing to immediately determine whether an estimator that attains the bound exists [14, p.27]. Notably the algorithm is used to examine the plausibility of the aforementioned partition of calculations. Most factors contributing to the performance or coverage of actual radar systems, such as antenna type or structure, form, size or previous data of the observed object, actual information carried by the signal, separate elements of the noise, and effects of the environment are ignored.

Implementation of a mathematical method to evaluate the theoretical lower bound for accuracy of unbiased estimator is deployed using the aforementioned algorithm to estimate *the mean squared error* (MSE) of desired parameters within the high SNR region. This method is then used to evaluate the effects of topological changes, since it yields the lower bound for expected error margins. The changes evaluated are addition and removal of an illuminator. These changes are then assimilated by combining data from other scenarios with the prediction before the change to achieve an acceptable prediction of the change. Calculations of the lower bound and the limiting volume simulations based on it through Monte Carlo integration are from now on referred as "the model". Expanding the model to incorrectly estimated situations as presented in [10] also provides a tool to evaluate accuracy and efficiency of real life estimators, but that is beyond the scope of this thesis.

2. VARIANCE ESTIMATION

A radar is an instrument that utilizes reflected electromagnetic radiation to estimate spatial attributes, such as location and velocity of objects. For any functioning radar, passive or active, the form of the primary signal is known or estimated and compared to the reflected secondary signal. Since electromagnetic radiation propagates in a multitude of ways in real life environment [26] and the intensity of any radiation is inversely proportionate to the distance squared, every radar has a theoretical maximum range of operation based on the antenna and signal analysis configuration. For a target to be made observable by reflecting electromagnetic signal, the secondary signal needs to be observed and identified from the surplus of electromagnetic radiation around. That is, the receiver needs sufficient signal-to-noise ratio from the target. The first concern is with deriving the spatial limit, at which the reflected SNR exceeds some threshold, and how those limits are affected by separating the calculation of different areas.

The basic tool for evaluating the SNR in the case of separate signal source and receiver is the bistatic radar equation [18, p.251].

$$\frac{P_r}{P_n} = \frac{P_t G_t}{4\pi d_i^2}\sigma_b \frac{1}{4\pi d_r^2}\frac{G_r \lambda^2}{4\pi}\frac{1}{k_B T_0 B F} L_s, \tag{2.1}$$

where P_r is received signal power, P_n is received noise power, P_t is transmit power, G_t is transmit antenna gain, d_i is illuminator-to-target range, σ_b is target bistatic radar cross-section, d_r is target-to-receiver range, G_r is receiver antenna gain, λ is the signal wavelength, k_B is the Boltzmann's constant, B is receiver effective bandwidth and F is receiver effective noise figure. In addition T_0 is the noise reference temperature, 290 K and $L_s(\leq 1)$ are the system losses. The components of the noise, over which the relevant signal must be observed, are listed in [18, p.254] and [26, p.34-1].

In addition to external noise, the scattering and reflecting of the radar signal, known as clutter, is a source of much dismay. Especially the effect of signal reflecting from the ground greatly affects the performance of monitoring objects close to the surface [23, p.23.22]. The internal noise and other disturbances induced by the

radar system itself are known as jitter [29, p.238]. Clutter may be incorporated to the SNR to form the *signal-to-clutter-plus-noise ratio* (SCNR). Other sources of clutter include a wide range of signal propagation, like reflection and absorption from distinct terrain structures, weather effects, and atmospheric diffraction and reflection [26, p.36-4].

Since variables d_i, d_r in Equation (2.1) are Euclidean distances and the other variables are assumed constant, on a two-dimensional plane the equation results in a quadratic plane curve, a *Cassini oval*, which indicates the area of sufficient SNR relative to transmitter and receiver configuration. However, it only yields elementary results for one-transmitter-one-receiver situation, as well as taking values like transmitter-to-target, target-to-receiver and receiver antenna gain as parameters; values which are substantially affected by wave propagation. It is therefore not directly applicable, when calculating actual radar coverage, but a useful tool for bounding the ideal estimation performance.

The actual parameters to solve the bistatic radar equation require specifying many variables, such as the target's radar cross-section, and thus all curves plotted would be antennae and target specific, the shape of the plane curve is parametrized by a non-negative scalar, the *Cassini parameter*. Let us define the parameter as the ratio of half the distance between the foci, and the square root of the product of distances from illuminator to target and receiver to target, such that

$$\frac{2\sqrt{d_i d_r}}{D} = \frac{b}{a}, \tag{2.2}$$

where b/a is the Cassini parameter defining the shape of the curve, and D is the distance between the illuminator and the receiver. The shape of the curves for some values of Cassini parameters are pictured in Figure 2.1.

The SNR and bistatic radar equation are functions of the physical parameters, and in no way estimate the effectiveness or accuracy of the actual object positioning. Although SNR can be used as an indicator of better performance with higher values, it lacks the capability to describe any object attribute excluding perhaps the effective radar cross-section. Thus a stricter bound is needed to estimate the relevant attributes of the target object. This is achieved by analysing the bound limited by the estimation algorithm.

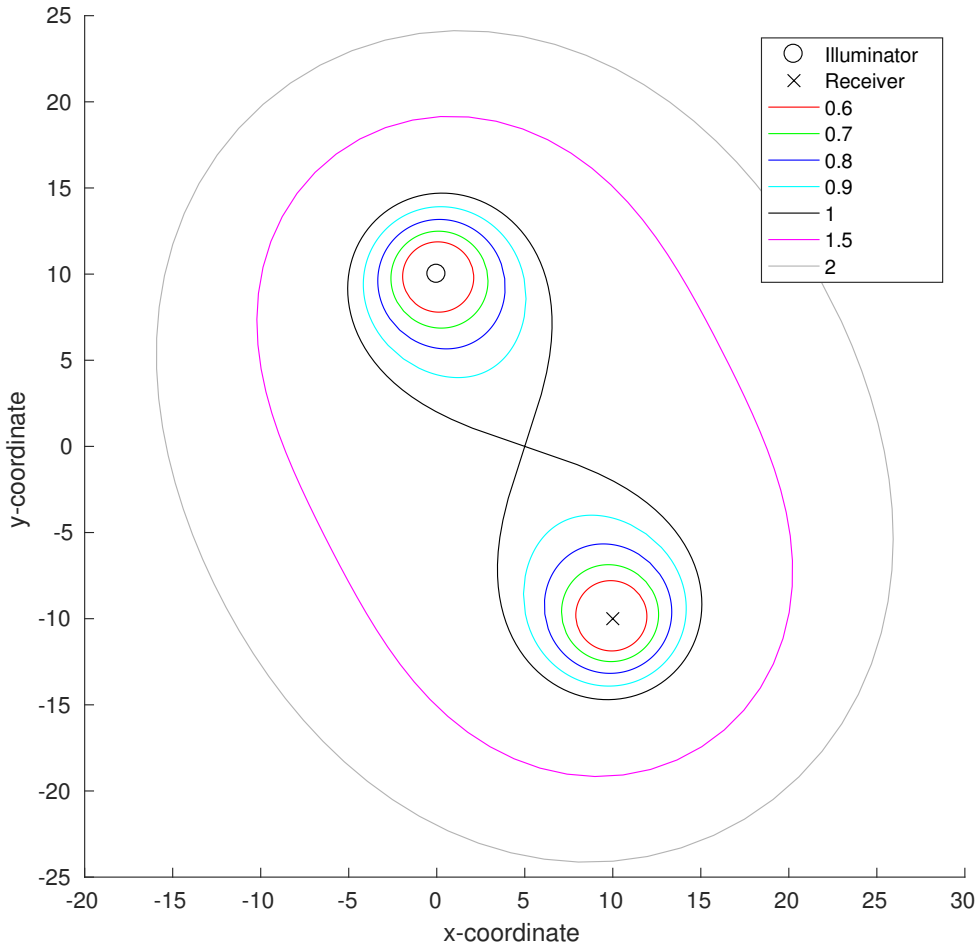

Figure 2.1 *SNR contour curves for different values of Cassini parameter*

2.1 Bayesian inference

Since the bistatic radar SNR in Equation (2.1) can not be used to estimate the accuracy of predictions, statistical tools, like Bayesian inference, are used to estimate the coverage limited by accuracy. Information about the population, here targets on a general level can be gathered from the data via induction. When more information is received from the population, that being the set of observed targets, the uncertainty of the knowledge decreases. This change in uncertainty is quantified by *inference*. [11, p.1-2]

Compared to common frequentist reasoning, the approach in Bayesian inference shifts the focus from determining the distribution of data to evaluating the accuracy of parameters that define the distribution of random variables at hand. Instead of referring to probabilities as objective properties of the real world, they are assumed

to describe the degree of belief. [28, p.175-176]

Definition 1. *A **random variable** is a mapping*

$$x : \Omega \to \mathbb{R} \tag{2.3}$$

that assigns a real value $x(\omega)$ to each outcome ω of sample space Ω. [28, p.19]

Discrete random variables are not of interest with respect to the underlying problem. Thus let us focus on *continuous random variables*. A random variable is continuous if there exists a function p_x such that $p_x(t) \geq 0$ for all $t \in \mathbb{R}$, $\int_{-\infty}^{\infty} p_x(t)dt = 1$ and for every $a \leq b$ the *probability measure*

$$\mathbb{P}(a < x < b) = \int_a^b p_x(t)dt \tag{2.4}$$

holds. The function p_x is called the *probability density function* (PDF) [28, p.23], and

$$P_x(a) = \int_{-\infty}^{a} p_x(t)dt \tag{2.5}$$

is the *cumulative distribution function*. The subscript x in p_x and P_x denotes the random variable over which the probability density is expressed. It is usually left out if there is no danger of misinterpretation. Inequality $a < x < b$ in Equation (2.4) represents an *event* and expresses the subset of outcomes in Ω that satisfy some arbitrary condition. We say that a continuous random variable is *normal* or *Gaussian distributed* if the probability density function is of the form [28, p.28]

$$p\left(x\right) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}\left(x-\mu\right)^2\right), \tag{2.6}$$

where $\mu \in \mathbb{R}$ is the mean value of the distribution and $\sigma^2 > 0$ is the variance. The distribution is denoted as $\mathcal{N}(\mu, \sigma^2)$, and a variable x being Gaussian distributed is denoted as $x \sim \mathcal{N}(\mu, \sigma^2)$ [28, p.28].

Definition 2. *A **random vector** is a multivariate random variable $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$, whose elements are random variables. The superscript $(\bullet)^T$ denotes matrix transpose.*

Similarly to univariate case a random vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$ is *continuous* if there exists probability density function $p(\boldsymbol{x})$ such that the n-fold integrals [17, p.26]

$$P(\boldsymbol{x}) = \int_{-\infty}^{x_1} \dots \int_{-\infty}^{x_n} p\left(\boldsymbol{t}\right) dt_1 \dots dt_n = \int_{-\infty}^{\boldsymbol{x}} p\left(\boldsymbol{t}\right) d\boldsymbol{t}, \tag{2.7}$$

and

$$\int_{-\infty}^{\infty} p\left(\boldsymbol{t}\right) d\boldsymbol{t} = 1 \tag{2.8}$$

hold. The cumulative distribution function is now defined element-wise as [17, p.26]

$$P_{\boldsymbol{x}}(\boldsymbol{u}) = \mathbb{P}\left(\boldsymbol{x} \leq \boldsymbol{u}\right) = \mathbb{P}\left(x_1 \leq u_1, x_2 \leq u_2, \dots, x_n \leq u_n\right), \tag{2.9}$$

and is equal to the joint probability of events $x_i \leq u_i$. Now the probability measure for $\boldsymbol{x}$ in any set $V \subseteq \mathbb{R}^n$ is [17, p.26]

$$\mathbb{P}\left(\boldsymbol{x} \in V\right) = \int_V p(\boldsymbol{t}) d\boldsymbol{t}. \tag{2.10}$$

The random vector $\boldsymbol{x}$ is said to have n-dimensional multinormal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ if its PDF is of the form [17, p.37]

$$p\left(\boldsymbol{x}\right) = \frac{1}{\sqrt{\mathrm{Det}(2\pi\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)^T \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)\right). \tag{2.11}$$

A special case of multivariate Gaussian variable is the *complex Gaussian variable* $z = x + iy \in \mathbb{C}$, where $i = \sqrt{-1}$. It combines two bivariate Gaussian distributed variables that express the real and imaginary parts respectively. A multivariate complex Gaussian is a collection of bivariate pairs, such that an n-variate complex Gaussian random variable is the vector $\boldsymbol{z} = [z_1, z_2, \dots, z_n]^T$ of bivariate real and imaginary parts $[x_1, y_1, x_2, y_2, \dots, x_n, y_n]^T$. [9, p.153] If the real and imaginary parts of a complex random variable are statistically independent and Gaussian with zero mean, the variable is said to be *circularly-symmetric* complex Gaussian [8]. From here on all complex Gaussian variables are assumed circularly-symmetric.

The PDF of a random variable indicates the relative likelihood within the probability distribution. Given events $A, B \subseteq \Omega$, and a probability measure $\mathbb{P} : \Omega \to [0, 1]$, the joint probability of the event "A and B", denoted as $A \wedge B$, is the probability, that both events "A" and "B" are true. For independent random variables it is defined as

$$\mathbb{P}(A \wedge B) = \mathbb{P}(A)\mathbb{P}(B).$$

For dependent case the joint probability is described by the *covariance* of the variables, which is a function of the respective multivariate distribution. The *conditional probability* of "A given B" clearly requires both A and B, and thus $A \wedge B$. Now by limiting the sample space from Ω to only those where B is satisfied, gives an equation of form

$$\frac{\mathbb{P}(A \wedge B)}{\mathbb{P}(B)} = \mathbb{P}(A|B),$$

which is in accordance to the chain rule of probability, where $A|B$ denotes the event "A given B". For dependent variables in random vector $\boldsymbol{x} = \left[\boldsymbol{x}_1^T, \boldsymbol{x}_2^T\right]^T = [x_1, x_2, \ldots, x_n]^T$ the conditional probability where $\boldsymbol{x}_1 = \boldsymbol{x}_{1_0}$ is some arbitrary fixed value, can be expressed with the conditional PDF [17, p.27]

$$p_{\boldsymbol{x}}\left(\boldsymbol{x}_2|\boldsymbol{x}_1 = \boldsymbol{x}_{1_0}\right) = \frac{p_{\boldsymbol{x}}\left(\boldsymbol{x}_{1_0}, \boldsymbol{x}_2\right)}{p_{\boldsymbol{x}_1}\left(\boldsymbol{x}_1\right)}, \tag{2.12}$$

where

$$p_{\boldsymbol{x}_1}\left(\boldsymbol{x}_1\right) = \int_{-\infty}^{\infty} p_{\boldsymbol{x}}\left(\boldsymbol{x}_1, \boldsymbol{x}_2\right) d\boldsymbol{x}_2 \tag{2.13}$$

is the *marginal* PDF of $\boldsymbol{x}_1$ and the integral notation is like in Equation (2.7).

Since the joint operation is commutative assuming $\mathbb{P}(A)$ is non-zero for inference to be meaningful, basic algebraic operations result in

$$\begin{aligned} \mathbb{P}(A|B)\mathbb{P}(B) &= \mathbb{P}(B|A)\mathbb{P}(A) \\ \Rightarrow \mathbb{P}(B|A) &= \frac{\mathbb{P}(A|B)}{\mathbb{P}(A)}\mathbb{P}(B) \end{aligned} \tag{2.14}$$

which is known as the *Bayes' theorem*. This gives a method for simple statistical inference: updating the degree of belief that $\mathbb{P}(B|A)$ based on the evidence on A. Assume there are samples from two separate sample spaces Ω and Θ, such that $\theta \in \Theta$ denotes some parameter describing the distribution of arbitrarily many data samples $\boldsymbol{x} = [x(\omega_1), x(\omega_2), \ldots, x(\omega_n)], \omega_i \in \Omega$, and Θ is a *parameter space*. Now $p(\boldsymbol{x}|\theta)$ is the PDF of $\boldsymbol{x}$ given some fixed parameter value $\theta = \theta_0$. If the PDF $p(\boldsymbol{x}|\theta)$ conditioned under θ has partial derivatives up to third order with respect to θ that are bound by integrable functions, and the range of $x(\omega_i) \in \mathbb{R}$ is independent of θ, then the probability density function is *regular*. [3, p.51] This applies whether θ is a single parameter or vector of parameters. The PDF is assumed regular unless stated otherwise.

Since the PDF describes the relative likelihood of an event in the distribution, it can substitute for the probabilities in Equation (2.14). If there is no information on the parameter, the distribution of $\boldsymbol{x}$ is the distribution over all possible values of θ, and can therefore be written as an integral over the parameter space

$$p(\boldsymbol{x}) = \int_{\Theta} p(\boldsymbol{x}|\theta)p(\theta)d\theta. \tag{2.15}$$

Replacing the denominator in Equation (2.14) with the previous integral yields

$$p(\theta|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\theta)p(\theta)}{\int_{\Theta} p(\boldsymbol{x}|\theta)p(\theta)d\theta}, \tag{2.16}$$

which is known as the *Bayes' rule* [11, p.2]. This is the basis of *Bayesian inference*.

Inference is a way to deduce information about the distribution of data by analysing the given data set. In Equation (2.16) the term $p(\theta)$ is knowledge, the best estimation of the PDF of $\boldsymbol{x}$ called the *prior distribution*. The estimation can be corrected and uncertainty lessened with data by using the Bayes' rule to deduce $p(\theta|\boldsymbol{x})$, the *posterior distribution*, which is the best estimate distribution of θ after observing some event that is mapped as $\boldsymbol{x}$ [11, p.1-2], which is the set of the data.

In Equation (2.16) the term $p(\boldsymbol{x}|\theta) = \mathcal{L}_n(\theta)$ is the *likelihood function*, and the logarithmic form of said function $L_n(\theta) = \log \mathcal{L}_n(\theta)$ is the *log-likelihood function*. The *maximum likelihood estimator* is defined as the value of θ that maximizes $\mathcal{L}_n(\theta)$, denoted by $\hat{\theta}_n$. [28, p.122] Although here the likelihood function is equal to probability density function of the data, it should not in general be confused with PDF in the sense that it represents the resemblance of the data in regard to the estimated distribution. It is treated as a function on the parameter θ and in general $\int_{\Theta} \mathcal{L}(\theta)d\theta \neq 1$ [28, p.122].

The *estimator* $\hat{\theta} = \theta(\boldsymbol{x})$, $\hat{\theta} : \mathbb{R} \to \Theta$ describes the variable θ based on the data. A good, but not essential property of an estimator is *unbiasedness*. An estimation given by an unbiased estimator is the actual value of the parameter [28, p.90], or

$$B(\hat{\theta}) = \langle \hat{\theta} - \theta \rangle = 0, \tag{2.17}$$

where $\langle \bullet \rangle$ denotes the expected value. The bias $B(\hat{\theta})$ squared is the *mean squared error* of the estimator $\hat{\theta}$

$$\text{MSE} = \langle \left(\hat{\theta} - \theta\right)^2 \rangle, \tag{2.18}$$

that measures the quality of the estimator. Notably the estimator is arbitrary,

and needs to be specified within problem, and the biasedness is a property of the estimator $\hat{\theta}$ only, not θ. A more reasonable requirement for an estimator is *consistency*: as evidence is presented, the estimator converges to the true parameter, $\hat{\theta} \xrightarrow{p} \theta$ [28, p.90], meaning that it is possible to make the number of observations sufficiently large for the probability of the estimation deviating from the true value less than some specified amount to become arbitrarily close to one. [3, p.48] This is not the most general and rigorously proved definition of the consistency, which assumes observations to be independent and identically distributed around their expectations, among a number of further conditions, which are demanding or difficult to verify in practice. [3, p.106] However, according to van den Bos "it is generally agreed in the literature that although not all maximum likelihood estimators are consistent, the conditions under which they are consistent are very general" [3, p.106], and this definition with convergence in probability is commonly suitable.

For a parameter θ, which could as well be a vector of variables, to define a distribution, a decision on a distribution model M has to be made. The distribution model defines how the distribution will be parametrized, and thus affects the size of the parameter vector θ. With Bayes' rule all operands were introduced with no regard to the underlying model of distribution. Including the model, affecting all operands, Equation (2.16) can be completed to

$$p(\theta|\boldsymbol{x}, M) = \frac{p(\boldsymbol{x}|\theta, M)p(\theta|M)}{p(\boldsymbol{x}|M)}, \tag{2.19}$$

which concludes the general case for Bayesian inference.

2.2 Fisher information

Fisher information measures the ability of the data to estimate a parameter, and the level of disorder of a system [7, p.23].

Definition 3. *A **Fisher information** of data set $\boldsymbol{x}$ about the parameter θ is the scalar*

$$I(\theta) = \left\langle \left(\frac{\partial}{\partial \theta} \log p(\boldsymbol{x}|\theta) \right)^2 \right\rangle_\theta \tag{2.20}$$

where $\left\langle \frac{\partial}{\partial \theta} \log p(\boldsymbol{x}|\theta) \right\rangle_\theta$ is the expected value of the rate at which the density $p(\boldsymbol{x}|\theta)$ changes at $\boldsymbol{x}$, while θ is constant. [15, p.115]

In Equation (2.20) the partial derivative of the logarithmic likelihood function is the

Fisher score function

$$S(\boldsymbol{x}|\theta) = \frac{\partial}{\partial\theta}\log p(\boldsymbol{x}|\theta), \tag{2.21}$$

which, being the gradient of logarithmic likelihood, indicates the sensitivity of the dependency between the likelihood function and θ: the higher the expected value of $\langle S(\boldsymbol{x}|\theta)\rangle_\theta$ is for any θ_0, the easier it is to distinguish θ_0 from any other θ [15, p.115].

The generalized vector extension of Equation (2.20) [15, p.125]

$$\boldsymbol{I}(\boldsymbol{\theta}) = \left[\boldsymbol{I}_{ij}(\boldsymbol{\theta})\right], \tag{2.22}$$

is the $n \times n$ square *Fisher information matrix* (FIM), where

$$\boldsymbol{I}_{ij}(\boldsymbol{\theta}) = \left\langle \frac{\partial}{\partial\theta_i}\log p(\boldsymbol{x}|\boldsymbol{\theta})\frac{\partial}{\partial\theta_j}\log p(\boldsymbol{x}|\boldsymbol{\theta})\right\rangle_{\boldsymbol{\theta}}. \tag{2.23}$$

The $n \times 1$ *Fisher score vector* for a vector of parameters $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_n]$ is defined similarly to the scalar case as

$$S(\boldsymbol{x}|\boldsymbol{\theta}) = \frac{\partial}{\partial\boldsymbol{\theta}}\log p(\boldsymbol{x}|\boldsymbol{\theta}). \text{ [3, p.23]} \tag{2.24}$$

Given that for any $\boldsymbol{x} \subset \Omega$, $\boldsymbol{\theta} \subset \Theta$, and $i = 1, \ldots, n$ the derivative $\frac{\partial p}{\partial\boldsymbol{\theta}}$ exists and is finite, as shown in [15, p.124-125],

$$\left\langle \frac{\partial}{\partial\theta_i}\log p(\boldsymbol{x}|\boldsymbol{\theta})\right\rangle_{\boldsymbol{\theta}} = 0$$

and

$$\boldsymbol{I}_{ij}(\boldsymbol{\theta}) = \operatorname{cov}\left(\frac{\partial}{\partial\theta_i}\log p(\boldsymbol{x}|\boldsymbol{\theta}), \frac{\partial}{\partial\theta_j}\log p(\boldsymbol{x}|\boldsymbol{\theta})\right). \text{ [15, p.125]} \tag{2.25}$$

Being a covariance matrix, $\boldsymbol{I}(\boldsymbol{\theta})$ is *positive semidefinite* and *positive definite* unless $\frac{\partial}{\partial\theta_i}\log p(\boldsymbol{x}|\boldsymbol{\theta})$, $i = 1, \ldots, n$ are linearly dependent. A positive semidefinite $n \times n$ square matrix A satisfies

$$\boldsymbol{a}^T A\boldsymbol{a} \geq 0$$

for any real $n \times 1$ vector $\boldsymbol{a}$. If the corresponding strict inequality holds for any real non-zero $n \times 1$ vector the matrix A is called positive definite. [3, p.259] The notation

$A \succeq B$ indicates that the matrix $A - B$ is positive semidefinite, which is a partial ordering on square matrices. An alternative expression for $\boldsymbol{I}_{ij}(\boldsymbol{\theta})$ is

$$\boldsymbol{I}_{ij}(\boldsymbol{\theta}) = -\left\langle \frac{\partial^2}{\partial\theta_i \partial\theta_j} \log p(\boldsymbol{x}|\boldsymbol{\theta}) \right\rangle_{\boldsymbol{\theta}}. \tag{2.26}$$

2.3 Cramér–Rao lower bound

Cramér-Rao lower bound can be used to measure the accuracy of an estimator even when the value of the estimated parameter is not known precisely. [10]. For the general class of unbiased estimators it defines a lower bound for the variance of the estimators.

Theorem 1. *Let $\hat{\theta}(\boldsymbol{x}) = \hat{\theta}$ be an unbiased scalar estimator of $\rho(\theta) = \rho$, $p(\boldsymbol{x}|\theta)$ the regular joint probability density function of available observations $\boldsymbol{x}$, and $\boldsymbol{I}_\theta$ the Fisher information $\boldsymbol{I}(\theta|\boldsymbol{x})$. [3, p.60] The* Cramér-Rao inequality *[3, p.60] states that the inequality*

$$\operatorname{var}\left(\hat{\theta}\right) \geq \frac{1}{\boldsymbol{I}_\theta}\left(\frac{d\rho}{d\theta}\right)^2 \tag{2.27}$$

bounds the variance of $\hat{\theta}$.

In Theorem 1 the Fisher information is positive definite, and thus invertible, since no bound is defined unless there is some information of all the parameters estimated jointly, and by assumption the data $\boldsymbol{x}$ is available. Theorem 1 is a special case of covariance, since for scalar θ $\operatorname{cov}(\hat{\theta}, \hat{\theta}) = \operatorname{var}(\hat{\theta})$.

Theorem 2. *With the same assumptions as previously, but now $\boldsymbol{\theta}$ is a vector $[\theta_1, \theta_2, \ldots, \theta_K]^T$, and respectively $\boldsymbol{\rho} = [\rho_1(\boldsymbol{\theta}), \rho_2(\boldsymbol{\theta}), \ldots, \rho_L(\boldsymbol{\theta})]^T$, $\hat{\boldsymbol{\theta}}$ is an unbiased estimator of $\boldsymbol{\rho}$ and $\boldsymbol{I}_{\boldsymbol{\theta}}$ is the Fisher information matrix $\boldsymbol{I}(\boldsymbol{\theta}|\boldsymbol{x})$, the covariance matrix is bounded by the inequality*

$$\operatorname{cov}\left(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}\right) \succeq \frac{\partial\boldsymbol{\rho}}{\partial\boldsymbol{\theta}^T}\boldsymbol{I}_{\boldsymbol{\theta}}^{-1}\frac{\partial\boldsymbol{\rho}^T}{\partial\boldsymbol{\theta}}, \quad \textit{[3, p.63]} \tag{2.28}$$

where $\frac{\partial\boldsymbol{\rho}}{\partial\boldsymbol{\theta}^T}$ is the $L \times K$ Jacobian matrix of $\boldsymbol{\rho}$ with respect to $\boldsymbol{\theta}$.

Proof. The proof is taken from [3, p.63-64]. Given, that $\boldsymbol{x}$ follows a joint probability density function $p(\boldsymbol{x}|\boldsymbol{\theta})$, assume

$$\langle\hat{\boldsymbol{\theta}}\rangle = \boldsymbol{\rho}$$

with

$$\langle \hat{\boldsymbol{\theta}} \rangle = \int \hat{\boldsymbol{\theta}} p(\boldsymbol{x}|\boldsymbol{\theta}) d\boldsymbol{x}.$$

Thus

$$\frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{\theta}^T} = \frac{\partial}{\partial \boldsymbol{\theta}^T} \int \hat{\boldsymbol{\theta}} p(\boldsymbol{x}|\boldsymbol{\theta}) d\boldsymbol{x}. \tag{2.29}$$

Since $p(\boldsymbol{x}|\boldsymbol{\theta})$ is known to be regular in the sense defined in Section 2.1, the order of differentiation and integration in this expression may be interchanged. In addition, it should be noted that

$$\frac{\partial}{\partial \boldsymbol{\theta}^T} p(\boldsymbol{x}|\boldsymbol{\theta}) = \frac{\partial \log p(\boldsymbol{x}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} p\left(\boldsymbol{x}|\boldsymbol{\theta}\right), \tag{2.30}$$

and thus

$$\frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{\theta}^T} = \int \hat{\boldsymbol{\theta}} \frac{\partial \log p(\boldsymbol{x}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} p(\boldsymbol{x}|\boldsymbol{\theta}) d\boldsymbol{x} = \langle \hat{\boldsymbol{\theta}} S_{\boldsymbol{\theta}}^T \rangle, \tag{2.31}$$

where $S_{\boldsymbol{\theta}}$ is the Fisher score $S(\boldsymbol{x}|\boldsymbol{\theta})_{\boldsymbol{\theta}}$. The expectation operation is known to be linear [28, p.50] and the covariance matrix $\operatorname{cov}\left(\hat{\boldsymbol{\theta}}, S_{\boldsymbol{\theta}}\right)$ is by definition [3, p.39]

$$\operatorname{cov}\left(\hat{\boldsymbol{\theta}}, S_{\boldsymbol{\theta}}\right) \triangleq \langle (\hat{\boldsymbol{\theta}} - \langle \hat{\boldsymbol{\theta}} \rangle) \left(S_{\boldsymbol{\theta}} - \langle S_{\boldsymbol{\theta}} \rangle\right)^T \rangle = \langle \hat{\boldsymbol{\theta}} S_{\boldsymbol{\theta}}^T \rangle - \langle \hat{\boldsymbol{\theta}} \rangle \langle S_{\boldsymbol{\theta}} \rangle^T. \tag{2.32}$$

The relation $\triangleq$ denotes equality by definition. Since $\langle S_{\boldsymbol{\theta}} \rangle = 0$, Equation (2.31) is equal to

$$\frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{\theta}^T} = \operatorname{cov}\left(\hat{\boldsymbol{\theta}}, S_{\boldsymbol{\theta}}\right), \tag{2.33}$$

which is the $L \times K$ covariance matrix of $\hat{\boldsymbol{\theta}}$ and $S_{\boldsymbol{\theta}}$. Considering the $(L + K) \times 1$ vector

$$\begin{bmatrix} \hat{\boldsymbol{\theta}} \\ S_{\boldsymbol{\theta}} \end{bmatrix},$$

the covariance matrix of this vector is

$$\begin{bmatrix} \operatorname{cov}\left(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}\right) & \operatorname{cov}\left(\hat{\boldsymbol{\theta}}, S_{\boldsymbol{\theta}}\right) \\ \operatorname{cov}\left(S_{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}\right) & \operatorname{cov}\left(S_{\boldsymbol{\theta}}, S_{\boldsymbol{\theta}}\right) \end{bmatrix},$$

which is positive semidefinite and equal to

$$\begin{bmatrix} \operatorname{cov}\left(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}\right) & \frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{\theta}^T} \\ \frac{\partial \boldsymbol{\rho}^T}{\partial \boldsymbol{\theta}} & \boldsymbol{I}_{\boldsymbol{\theta}} \end{bmatrix}.$$

Suppose that $\boldsymbol{I}_{\boldsymbol{\theta}}$ is nonsingular, so that $\boldsymbol{I}_{\boldsymbol{\theta}}^{-1}$ exists. Consider the $(L+K) \times L$ matrix

$$\begin{bmatrix} I \\ -\boldsymbol{I}_{\boldsymbol{\theta}}^{-1} \frac{\partial \boldsymbol{\rho}^T}{\partial \boldsymbol{\theta}} \end{bmatrix},$$

where I is the identity.

Let $\boldsymbol{V}$ be real symmetric positive definite $M \times M$ matrix, and $\boldsymbol{P}$ be real $M \times N$ matrix. Now for any $N \times 1$ vector $\boldsymbol{a}$ the scalar

$$\boldsymbol{a}^T \boldsymbol{P}^T \boldsymbol{V} \boldsymbol{P} \boldsymbol{a} = \boldsymbol{b}^T \boldsymbol{V} \boldsymbol{b},$$

is positive, if $\boldsymbol{b} = \boldsymbol{P}\boldsymbol{a} \neq \boldsymbol{0}$ and zero if $\boldsymbol{b} = \boldsymbol{0}$. Since $\boldsymbol{P}$ may be singular and thus $\boldsymbol{P}\boldsymbol{a} = \boldsymbol{0}$ even if $\boldsymbol{a} \neq \boldsymbol{0}$, the matrix $\boldsymbol{P}^T \boldsymbol{V} \boldsymbol{P}$ is positive semidefinite. [3, p.261-262] Therefore the product

$$\begin{bmatrix} I & -\frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{\theta}^T} \boldsymbol{I}_{\boldsymbol{\theta}}^{-1} \end{bmatrix} \begin{bmatrix} \operatorname{cov}\left(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}\right) & \frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{\theta}^T} \\ \frac{\partial \boldsymbol{\rho}^T}{\partial \boldsymbol{\theta}} & \boldsymbol{I}_{\boldsymbol{\theta}} \end{bmatrix} \begin{bmatrix} I \\ -\boldsymbol{I}_{\boldsymbol{\theta}}^{-1} \frac{\partial \boldsymbol{\rho}^T}{\partial \boldsymbol{\theta}} \end{bmatrix}, \tag{2.34}$$

is positive semidefinite since it is of the form $\boldsymbol{P}^T \boldsymbol{V} \boldsymbol{P}$ with $\boldsymbol{V}$ being positive semidefinite. Partitioned multiplication, meaning the multiplications is done in a familiar manner, but regarding the block matrices as elements of appropriate size of the matrix, in Equation (2.34) produces the matrix

$$\operatorname{cov}\left(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}\right) - \frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{\theta}^T} \boldsymbol{I}_{\boldsymbol{\theta}}^{-1} \frac{\partial \boldsymbol{\rho}^T}{\partial \boldsymbol{\theta}}, \tag{2.35}$$

which, being positive semidefinite, completes the proof. [3, p.63-64] □

The matrix

$$\frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{\theta}^T} \boldsymbol{I}_{\boldsymbol{\theta}}^{-1} \frac{\partial \boldsymbol{\rho}^T}{\partial \boldsymbol{\theta}} \tag{2.36}$$

from Equation (2.28) is called the *Cramér-Rao lower bound matrix*, with the variances of $\rho_l(\boldsymbol{\theta})$ as diagonal elements [3, p.63]. If the function $\rho(\boldsymbol{\theta})$ is used to estimate

the $K \times 1$ parameter vector $\boldsymbol{\theta}$ itself, then $L = 1$ and the Jacobian differential [3, p.65]

$$\frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{\theta}^T} = I,$$

and

$$\operatorname{cov}\left(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}\right) \succeq \boldsymbol{I}_\theta^{-1}, \tag{2.37}$$

so for estimating the parameter the covariance matrix of the estimator has a lower bound: the inverse of the Fisher information matrix. This concurs with the inequality

$$\boldsymbol{\sigma}_i^2 \boldsymbol{I}_{\theta,ii} \triangleq \operatorname{var}(\theta_i) \boldsymbol{I}_{\theta,ii} \geq 1, \tag{2.38}$$

which states that for any unbiased estimate of $\boldsymbol{\theta}$, the MSE of θ_i is equal or greater than the ith diagonal of the inverse of the Fisher information matrix on $\boldsymbol{\theta}$ [27, p.258].

3. CRLB FOR JOINT VELOCITY AND POSITION ESTIMATION

The variation of coverage induced by topological changes do not require re-evaluation of the full coverage, if it as a whole can be interpreted as a union of illuminator-receiver pairs. With free path propagation the hard SNR bound is rather effortlessly achieved by combining the single Cassini ovals, with loci at a single illuminator and a single receiver, as shown in Chan 2008 [4]. In actual use cases however, the free path Cassini oval is seldom achieved as the SNR limit without simulation or actual measurement data, since the receiver effective noise figure and system losses are assumably difficult to predict, and antenna orientation and atmospheric and ground propagation result in non-analytical contour curve. In addition, reflected signals beyond a well defined SNR range could achieve wanted accuracy in certain regions, where multiple weak signals complement each other. Removing one illuminator-receiver pair from the total coverage may affect areas within the overlap, or even outside of the ovals. However, when evaluating the best possible accuracy, the undesired effects of presuming total coverage as a union of ovals may span additional miscalculations to the model.

Let us decide on a minimal allowed accuracy for the observations in theoretical maximum range, say, a target with expected reflective cross-section of σ must be located within specified bound of the calculated result with the probability P. Complex Bayesian statistical methods can be used to calculate a lower bound of the variance, which is related to the error. Conversely it is a upper bound for the accuracy. If the highest possible accuracy is unacceptably low within assumed or previously computed coverage, it limits the coverage area further.

The inverse of Fisher information matrix of a vector parameter is the Cramér-Rao lower bound for the estimated covariance matrix, like shown in Equation (2.37), assuming an unbiased estimator. The following algorithm gives the standard CRLB for the mean squared error of an arbitrary vector parameter [10], and thus provides a great reference for evaluating the best case scenario.

The generalised Cramér-Rao lower bound for system with an arbitrary number of

illuminators and receivers, that gives joint estimation on target velocity and position is provided by He *et al.* [10]. Next the aforementioned lower bound is derived closely following the example set in [10].

Let

$$\boldsymbol{\theta} = [x, y, z, v_x, v_y, v_z]^T = \left[\boldsymbol{p}^T, \boldsymbol{v}^T\right]^T \tag{3.1}$$

represent the vector of spatial parameters of the target to be estimated with $\boldsymbol{p} = [x, y, z]^T$ and $\boldsymbol{v} = [v_x, v_y, v_z]^T$, and

$$\boldsymbol{p}_m^i = \left[x_m^i, y_m^i, z_m^i\right]^T, m = 1, \ldots, M \tag{3.2}$$

$$\boldsymbol{p}_n^r = [x_n^r, y_n^r, z_n^r]^T, n = 1, \ldots, N \tag{3.3}$$

the coordinates of M widely spaced single antenna transmit stations: illuminators, and the N widely spaced single antenna receivers, respectively, in a three-dimensional Cartesian coordinate system. Using well-known geometry, one gets

$$d_{im} = \|\boldsymbol{p}_m^i - \boldsymbol{p}\|, \tag{3.4}$$

which is the Euclidean distance between the target and the mth transmitter, and similarly

$$d_{rn} = \|\boldsymbol{p}_n^r - \boldsymbol{p}\|, \tag{3.5}$$

which is the distance between the target and the nth receiver. The Doppler shift

$$f_{nm} = \frac{\boldsymbol{v} \cdot (\boldsymbol{p}_m^i - \boldsymbol{p})}{\lambda d_{im}} + \frac{\boldsymbol{v} \cdot (\boldsymbol{p}_n^r - \boldsymbol{p})}{\lambda d_{rn}}, \tag{3.6}$$

as well as the time delay

$$\tau_{nm} = \frac{d_{im} + d_{rn}}{c} \tag{3.7}$$

are formed in a similar fashion, where λ denotes wavelength and c the speed of light in vacuum. With these spatial variables the received waveform at the nth receiver at time kT_s, $k = 1, \ldots, K$ being the index over time samples, and T_s the sampling period, is

$$r_n(k) = \sum_{m=1}^{M} \sqrt{\frac{E_m P_0}{d_{im}^2 d_{rm}^2}} \zeta_{nm} s_m(kT_s - \tau_{nm}, \boldsymbol{\alpha}_m) \exp(j2\pi f_{nm} k T_s) + w_n(k), \tag{3.8}$$

where $\boldsymbol{\zeta} = [\zeta_{11}, \zeta_{12}, \ldots, \zeta_{NM}]^T$ represents the reflection coefficient vector, and ζ_{nm} corresponds to the nmth path. The reflection coefficient is assumed to be constant over the observation interval and to have a known complex, in other words bivariate, Gaussian statistics model with zero mean and covariance matrix $\boldsymbol{R} = \langle \boldsymbol{\zeta}\boldsymbol{\zeta}^H \rangle \in \mathbb{C}^{NM \times NM}$, that is $\boldsymbol{\zeta} \sim \mathcal{CN}(\boldsymbol{0}, \boldsymbol{R})$. The superscript $(\bullet)^H$ indicates conjugate transpose of a matrix. The product $\sqrt{E_m} s_m(k, \boldsymbol{\alpha}_m)$ is the full time-sampled version of the signal transmitted from the mth transmit station at time instant kT_s, where $k = 1, \ldots, K$ is an index running over the different time samples. Parameter vector $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_M]^T$ is needed to describe the waveform. The vector $\boldsymbol{w} = [w_1, w_2, \ldots, w_N]^T$ represents the clutter-plus-noise vectors, where $w_n = [w_n(1), w_n(2), \ldots, w_n(K)]^T$ is a vector containing the noise samples from the nth receiver. The vectors are assumed to be complex Gaussian distributed with zero mean and covariance matrix $\boldsymbol{Q} = \langle \boldsymbol{w}\boldsymbol{w}^H \rangle \in \mathbb{C}^{NK \times NK}$, that is $\boldsymbol{w} \sim \mathcal{CN}(\boldsymbol{0}, \boldsymbol{Q})$. The noise vector $\boldsymbol{w}$ is assumed to be independent from the reflection coefficient vector $\boldsymbol{\zeta}$. $\sum_{k=1}^{K} |s_m(k, \boldsymbol{\alpha}_m)|^2 T_s = 1$ is the normalization used on the waveform. The received signal strength at $d_{im} = d_{rn} = 1$ is $\sqrt{E_m P_0}$, so P_0 denotes the ratio of received energy at $d_{im} = d_{rn} = 1$ to transmitted energy. [10]

The observations from the nth receiver can be expressed as

$$\begin{aligned} \boldsymbol{r}_n &= [r_n(1), r_n(2), \ldots, r_n(K)]^T \\ &= \boldsymbol{U}_n \boldsymbol{\zeta}_n + \boldsymbol{w}_n \qquad (3.9) \\ \boldsymbol{r} &= \boldsymbol{S}\boldsymbol{\zeta} + \boldsymbol{w}, \qquad (3.10) \end{aligned}$$

where $\boldsymbol{U}_n$ is the $K \times M$ matrix that collects the time delay and Doppler shifted signals at the nth receiver as

$$\boldsymbol{U}_n = [\boldsymbol{u}_n(1), \boldsymbol{u}_n(2), \ldots, \boldsymbol{u}_n(K)]^T, \qquad (3.11)$$

where

$$\boldsymbol{u}_n(k) = [u_{n1}(1), u_{n2}(2), \ldots, u_{nM}(K)]^T,$$

and

$$u_{nm}(k) = \sqrt{\frac{E_m P_0}{d_{tm}^2 d_{rm}^2}} s_m(kT_s - \tau_{nm}, \boldsymbol{\alpha}_m) \exp(j2\pi f_{nm} k T_s). \qquad (3.12)$$

The matrix $\boldsymbol{S}$ is the $N * K \times N * M$ diagonal block matrix

$$\boldsymbol{S} = \mathrm{Diag}\left(\boldsymbol{U}_1, \boldsymbol{U}_2, \ldots, \boldsymbol{U}_N\right) \tag{3.13}$$

with the signal matrices $\boldsymbol{U_n}$ as diagonal elements. Following He *et al.* [10] and assuming $\boldsymbol{S}$ (and thus $\boldsymbol{\alpha}$), $\boldsymbol{Q}$ and $\boldsymbol{R}$ are known to the estimation algorithm and knowing that linear combination of two Gaussian vectors is Gaussian, the likelihood function is

$$\mathcal{L}\left(\boldsymbol{r}|\boldsymbol{\theta}, \boldsymbol{\alpha}\right) = \frac{1}{\pi^{KN}\,\mathrm{Det}\left(\boldsymbol{C}\right)} \exp\left(-\boldsymbol{r}^H \boldsymbol{C}^{-1} \boldsymbol{r}\right) \tag{3.14}$$

conditioned on $\boldsymbol{\alpha}$, and thus the log-likelihood function is

$$\begin{aligned} L\left(\boldsymbol{r}|\boldsymbol{\theta}, \boldsymbol{\alpha}\right) &= \ln \mathcal{L}\left(\boldsymbol{r}|\boldsymbol{\theta}, \boldsymbol{\alpha}\right) \\ &= -\boldsymbol{r}^H \boldsymbol{C}^{-1} \boldsymbol{r} - \ln\left(\mathrm{Det}\left(\boldsymbol{C}\right)\right) - KN \ln \pi, \end{aligned} \tag{3.15}$$

where $\boldsymbol{C}$ denotes the covariance matrix

$$\begin{aligned} \boldsymbol{C} &= \left\langle \left(\boldsymbol{S\zeta} + \boldsymbol{w}\right) \left(\boldsymbol{S\zeta} + \boldsymbol{w}\right)^H \right\rangle \\ &= \left\langle \boldsymbol{S\zeta\zeta}^H \boldsymbol{S}^H + \boldsymbol{ww}^H \right\rangle \\ &= \boldsymbol{SRS}^H + \boldsymbol{Q}. \end{aligned} \tag{3.16}$$

By omitting the last term of Equation 3.16, which is a constant, a maximum likelihood estimate of the unknown parameter vector $\boldsymbol{\theta}$ can be written as

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{ML} &= \arg\max_{\boldsymbol{\theta}} L\left(\boldsymbol{r}|\boldsymbol{\theta}, \boldsymbol{\alpha}\right) \\ &= \arg\max_{\boldsymbol{\theta}} \{-\boldsymbol{r}^H \boldsymbol{C}^{-1} \boldsymbol{r} - \ln\left(\mathrm{Det}\left(\boldsymbol{C}\right)\right)\} \end{aligned} \tag{3.17}$$

It is known from Equation (2.37), that the CRLB is the inverse of Fisher information matrix. Based on the model specified earlier in this section the FIM is a 6×6 matrix of second derivatives of the log-likelihood function, of the form

$$\boldsymbol{I}\left(\boldsymbol{\theta}|\boldsymbol{\alpha}\right) = \left\langle \nabla_{\boldsymbol{\theta}} L\left(\boldsymbol{r}|\boldsymbol{\theta}, \boldsymbol{\alpha}\right) \left(\nabla_{\boldsymbol{\theta}} L\left(\boldsymbol{r}|\boldsymbol{\theta}, \boldsymbol{\alpha}\right)\right)^T \right\rangle_{\boldsymbol{\theta}, \boldsymbol{\alpha}}. \tag{3.18}$$

Let us define an intermediate parameter vector

$$\begin{aligned}\boldsymbol{\vartheta} &= \left[\boldsymbol{\tau}^T, \boldsymbol{f}^T, \boldsymbol{d}_i^T, \boldsymbol{d}_r^T\right] \\ &= [\tau_{11}, \tau_{12}, \ldots, \tau_{NM}, f_{11}, f_{12}, \ldots, f_{NM}, \\ &\quad d_{i1}, d_{i2}, \ldots, d_{iM}, d_{r1}, d_{r1}, \ldots, d_{rN}]\end{aligned} \tag{3.19}$$

containing the parameters τ_{nm}, f_{nm}, d_{im} and d_{rn} of the likelihood function, where $n = 1, \ldots, N$ and $m = 1, \ldots, M$. The mathematical model contains symmetry uncertainties emerging from the system being underdetermined with respect to six spatial values derived from four parameters. For instance, if the antennae are completely isotropic and only one pair of receiver and illuminator exists, the information whether a stationary target is below of above the level of the illuminator and the receiver is unknown. In the aforementioned situation the level can be arbitrarily chosen since the lower bound is circularly symmetric. These might occur with a small number of topological elements or if they do not span three dimensional space. However, in simulations where target attributes are known this is not an issue, and outside simulation such pathological configuration would be highly unlikely, and solvable in multitude of ways. The Fisher information matrix of the intermediate parameters can be formed similarly the actual FIM in Equation (3.18), and is therefore

$$\boldsymbol{I}\left(\boldsymbol{\vartheta}|\boldsymbol{\alpha}\right) = \left\langle \nabla_{\boldsymbol{\vartheta}} L\left(\boldsymbol{r}|\boldsymbol{\vartheta}, \boldsymbol{\alpha}\right) \left(\nabla_{\boldsymbol{\vartheta}} L\left(\boldsymbol{r}|\boldsymbol{\vartheta}, \boldsymbol{\alpha}\right)\right)^T \right\rangle_{\boldsymbol{\vartheta},\boldsymbol{\alpha}}. \tag{3.20}$$

Now $\boldsymbol{\vartheta}$ is a function of $\boldsymbol{\theta}$ and the wanted FIM can be expressed as a function composition $\boldsymbol{I}\left(\boldsymbol{\vartheta}\left(\boldsymbol{\theta}\right)|\boldsymbol{\alpha}\right)$. According to the chain rule of composite functions, which states that the derivative of a composite function is the product of the derivatives of the outer function with respect to the inner function and of the inner function with respect to the variable at hand, the desired FIM $\boldsymbol{I}\left(\boldsymbol{\theta}|\boldsymbol{\alpha}\right)$ can be expressed as

$$\boldsymbol{I}\left(\boldsymbol{\theta}|\boldsymbol{\alpha}\right) = \left(\nabla_{\boldsymbol{\theta}}\boldsymbol{\vartheta}^T\right) \boldsymbol{I}\left(\boldsymbol{\vartheta}|\boldsymbol{\alpha}\right) \left(\nabla_{\boldsymbol{\theta}}\boldsymbol{\vartheta}^T\right)^T, \tag{3.21}$$

where $\nabla_{\boldsymbol{\theta}}\boldsymbol{\vartheta}^T$ is the derivative of the inner function. The derivatives of the outer function are contained in $\boldsymbol{I}\left(\boldsymbol{\vartheta}|\boldsymbol{\alpha}\right)$. Now two terms need to be solved to achieve the general CRLB: the derivative matrix $\nabla_{\boldsymbol{\theta}}\boldsymbol{\vartheta}^T$ and the Fisher information matrix of intermediate parameters $\boldsymbol{I}\left(\boldsymbol{\vartheta}|\boldsymbol{\alpha}\right)$.

3.1 Term $\nabla_{\boldsymbol{\theta}}\boldsymbol{\vartheta}^T$

In Equation (3.1) the parameter variables are defined, over which the partial derivative of the intermediate parameter vector in Equation (3.19) is taken. Since Doppler

shifts are the only variables that depend on the velocity of the target, the partially differentiated intermediate parameter vector is of the form [10]

$$\nabla_{\boldsymbol{\theta}}\boldsymbol{\vartheta}^T = \begin{bmatrix} \boldsymbol{F} & \boldsymbol{G} & \boldsymbol{D}_i & \boldsymbol{D}_r \\ \boldsymbol{0} & \boldsymbol{H} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \tag{3.22}$$

where

$$\boldsymbol{F} = \frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{p}} = \begin{bmatrix} \frac{\partial \tau_{11}}{\partial x} & \frac{\partial \tau_{12}}{\partial x} & \cdots & \frac{\partial \tau_{NM}}{\partial x} \\ \frac{\partial \tau_{11}}{\partial y} & \frac{\partial \tau_{12}}{\partial y} & \cdots & \frac{\partial \tau_{NM}}{\partial y} \\ \frac{\partial \tau_{11}}{\partial z} & \frac{\partial \tau_{12}}{\partial z} & \cdots & \frac{\partial \tau_{NM}}{\partial z} \end{bmatrix}, \tag{3.23}$$

$$\boldsymbol{G}_{ij} = \frac{\partial f_j}{\partial \boldsymbol{p}_i} \tag{3.24}$$

and

$$\boldsymbol{H}_{ij} = \frac{\partial f_j}{\partial \boldsymbol{v}_i} \tag{3.25}$$

are $3 \times N * M$ block matrices, whereas

$$\boldsymbol{D}_i = \frac{\partial \boldsymbol{d}_i}{\partial \boldsymbol{p}} = \begin{bmatrix} \frac{\partial d_{i1}}{\partial x} & \frac{\partial d_{i2}}{\partial x} & \cdots & \frac{\partial d_{iM}}{\partial x} \\ \frac{\partial d_{i1}}{\partial y} & \frac{\partial d_{i2}}{\partial y} & \cdots & \frac{\partial d_{iM}}{\partial y} \\ \frac{\partial d_{i1}}{\partial z} & \frac{\partial d_{i2}}{\partial z} & \cdots & \frac{\partial d_{iM}}{\partial z} \end{bmatrix} \tag{3.26}$$

and

$$\boldsymbol{D}_{r_{ij}} = \frac{\partial d_{rj}}{\partial \boldsymbol{p}_i} \tag{3.27}$$

are $3 \times M$ and $3 \times N$ block matrices, respectively.

According to Equations (3.4) - (3.7) one can represent every element in the matrices, fifteen different types of partial derivatives as scalar valued functions of the position and velocity of the target, and the locations of illuminators and receivers like such [10]

$$\frac{\partial \tau_{nm}}{\partial \boldsymbol{p}_j} = \frac{1}{c}\left(\frac{\boldsymbol{p}_j - \boldsymbol{p}^i_{mj}}{\|\boldsymbol{p}^i_m - \boldsymbol{p}\|} + \frac{\boldsymbol{p}_j - \boldsymbol{p}^r_{nj}}{\|\boldsymbol{p}^r_n - \boldsymbol{p}\|} \right),$$

$$\begin{aligned}\frac{\partial f_{nm}}{\partial \boldsymbol{p}_j} = &-\frac{\boldsymbol{v}_j}{\lambda}\left(\frac{1}{\|\boldsymbol{p}_m^i - \boldsymbol{p}\|} + \frac{1}{\|\boldsymbol{p}_n^r - \boldsymbol{p}\|}\right) \\ &+\frac{(\boldsymbol{p}_{mj}^i - \boldsymbol{p}_j)}{\lambda\|\boldsymbol{p}_m^i - \boldsymbol{p}\|^3}\left[\boldsymbol{v}\cdot\left(\boldsymbol{p}_m^i - \boldsymbol{p}\right)\right] \\ &+\frac{(\boldsymbol{p}_{nj}^r - \boldsymbol{p}_j)}{\lambda\|\boldsymbol{p}_n^r - \boldsymbol{p}\|^3}\left[\boldsymbol{v}\cdot(\boldsymbol{p}_n^r - \boldsymbol{p})\right],\end{aligned}$$

$$\frac{\partial f_{nm}}{\partial \boldsymbol{v}_j} = \frac{\boldsymbol{p}_{mj}^i - \boldsymbol{p}_j}{\lambda\|\boldsymbol{p}_m^i - \boldsymbol{p}\|} + \frac{\boldsymbol{p}_{nj}^r - \boldsymbol{p}_j}{\lambda\|\boldsymbol{p}_n^r - \boldsymbol{p}\|},$$

$$\frac{\partial d_{im}}{\partial \boldsymbol{p}_j} = \frac{\boldsymbol{p}_j - \boldsymbol{p}_{mj}^i}{\|\boldsymbol{p}_m^i - \boldsymbol{p}\|},$$

and

$$\frac{\partial d_{rn}}{\partial \boldsymbol{p}_j} = \frac{\boldsymbol{p}_j - \boldsymbol{p}_{nj}^r}{\|\boldsymbol{p}_n^r - \boldsymbol{p}\|}.$$

Notably here the index j indicates the jth variable of vectors in Equations (3.2) and (3.3) instead of a column of the derivative matrix. All derivatives with respect to different axis have the following identifiers

$$\frac{\partial \tau_{nm}}{\partial \boldsymbol{p}} = \begin{bmatrix} a_{nm} \\ b_{nm} \\ c_{nm} \end{bmatrix}, \frac{\partial f_{nm}}{\partial \boldsymbol{p}} = \begin{bmatrix} e_{nm} \\ g_{nm} \\ h_{nm} \end{bmatrix}, \frac{\partial f_{nm}}{\partial \boldsymbol{v}} = \begin{bmatrix} \beta_{nm} \\ \kappa_{nm} \\ \mu_{nm} \end{bmatrix}, \frac{\partial d_{im}}{\partial \boldsymbol{p}} = \begin{bmatrix} \nu_{nm} \\ \iota_{nm} \\ \rho_{nm} \end{bmatrix}, \frac{\partial d_{rn}}{\partial \boldsymbol{p}} = \begin{bmatrix} \eta_{nm} \\ \psi_{nm} \\ \omega_{nm} \end{bmatrix},$$

which are used to derive FIM elements in Appendix A: Information matrix elements.

3.2 Term $\boldsymbol{I}(\boldsymbol{\vartheta}|\boldsymbol{\alpha})$

Since the covariance matrix $\boldsymbol{C}$ is known to be Gaussian, and the form of the log-likelihood function in Equation (3.15) is suitable, the identity [14, p.73]

$$\frac{\partial \ln \mathrm{Det}(\boldsymbol{C})}{\partial \boldsymbol{\vartheta}_i} = \mathrm{Tr}\left(\boldsymbol{C}^{-1}\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{\vartheta}_i}\right), \tag{3.28}$$

where Tr(•) indicates the sum of main diagonal elements, or the trace, applies. Like in [10], the ijth element of the $\boldsymbol{I}(\boldsymbol{\vartheta}|\boldsymbol{\alpha})$, which is twice differentiated, can be expressed as

$$[\boldsymbol{I}(\boldsymbol{\vartheta}|\boldsymbol{\alpha})]_{ij} = \mathrm{Tr}\left(\boldsymbol{C}^{-1}\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{\vartheta}_i}\boldsymbol{C}^{-1}\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{\vartheta}_j}\right). \tag{3.29}$$

Using the identities from [20]

$$\mathrm{Tr}(\boldsymbol{ABXY}) = \mathrm{vec}\left(\boldsymbol{Y}^T\right)^T\left(\boldsymbol{X}^T \otimes \boldsymbol{A}\right)\mathrm{vec}(\boldsymbol{B}), \tag{3.30}$$

where vec(•) is the column vectorizing operator, and

$$\mathrm{Tr}(\boldsymbol{AB}) = \mathrm{Tr}(\boldsymbol{BA}), \tag{3.31}$$

so Equation (3.29) can be rewritten as [10]

$$\begin{aligned}[\boldsymbol{I}(\boldsymbol{\vartheta}|\boldsymbol{\alpha})]_{ij} &= \mathrm{Tr}\left(\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{\vartheta}_i}\boldsymbol{C}^{-1}\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{\vartheta}_j}\boldsymbol{C}^{-1}\right) \\ &= \left(\frac{\partial \mathrm{vec}(\boldsymbol{C})}{\partial \boldsymbol{\vartheta}_i}\right)^H\left(\boldsymbol{C}^{-T} \otimes \boldsymbol{C}^{-1}\right)\left(\frac{\partial \mathrm{vec}(\boldsymbol{C})}{\partial \boldsymbol{\vartheta}_j}\right),\end{aligned} \tag{3.32}$$

where $\boldsymbol{C}$ and $\boldsymbol{\vartheta}$ are known. The elements of the resulting 4×4 block matrix are computed in Appendix B: MATLAB Scripts.

3.3 Fisher information matrix

Based on Sections 3.1 and 3.2 the final FIM of vector $\boldsymbol{\theta}$ for given $\boldsymbol{\alpha}$ can be derived as the sum

$$\boldsymbol{I}(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \sum_{p=1}^{N}\sum_{q=1}^{M}\sum_{n=1}^{N}\sum_{m=1}^{M}\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{16} \\ A_{21} & A_{22} & \cdots & A_{26} \\ \vdots & \vdots & \ddots & \vdots \\ A_{61} & A_{62} & \cdots & A_{66} \end{bmatrix}. \tag{3.33}$$

Elements A_{ij} of the matrix in Equation (3.33) can be presented with block matrices

of the FIM of $\boldsymbol{\vartheta}$ given $\boldsymbol{\theta}$. The block matrices of

$$\boldsymbol{I}\left(\boldsymbol{\vartheta}|\boldsymbol{\alpha}\right) = \begin{bmatrix} \boldsymbol{I}_{\tau\tau} & \boldsymbol{I}_{\tau f} & \boldsymbol{I}_{\tau d_t} & \boldsymbol{I}_{\tau d_r} \\ \boldsymbol{I}_{f\tau} & \boldsymbol{I}_{ff} & \boldsymbol{I}_{f d_t} & \boldsymbol{I}_{f d_r} \\ \boldsymbol{I}_{d_t\tau} & \boldsymbol{I}_{d_t f} & \boldsymbol{I}_{d_t d_t} & \boldsymbol{I}_{d_t d_r} \\ \boldsymbol{I}_{d_r\tau} & \boldsymbol{I}_{d_r f} & \boldsymbol{I}_{d_r d_t} & \boldsymbol{I}_{d_r d_r} \end{bmatrix}, \tag{3.34}$$

that are defined in Section 3.2 and derived in Appendix A of [10] for two dimensional case, and in Appendix B: MATLAB Scripts for three dimensional solutions. The scalar elements of the matrix in Equation (3.33) are presented in Appendix A: Information matrix elements.

The information matrix is a twofold sum over both, the illuminators and the receivers since the indices p and n iterate over all receivers and the indices q and m over all illuminators. However the block matrices of $\boldsymbol{I}\left(\boldsymbol{\vartheta}|\boldsymbol{\alpha}\right)$ depend on the system as a whole. In He *et al.* Appendix A [10] the sampled waveforms of signals from all illuminators and to all receivers are derivatised with respect to the intermediate parameters from Equation (3.19) to form the Fisher information matrix of the intermediate parameter vector $\boldsymbol{\vartheta}$. The element-wise solutions to the block matrices are highly convoluted, so analytical modularisation of the CRLB is not a part of this thesis.

4. SIMULATIONS

The coverage and performance of a radar system is implementation specific. The estimation algorithms used to derive the spatial parameters describing location and movement of target dictates the true accuracy of the system. Bias is the difference between the expected value of the estimator and the hypothetical true value of the parameter [3, p.2]. The results in this thesis assume one receiver and a perfect estimator, which eliminates estimator bias.

The CRLB is calculated by Matlab using Moore-Penrose pseudoinverse of FIM, since low information in distant coordinates may result in close to singular matrices. This is not seen as problem, since the data seems coherent over all simulations, and clearly non-singular matrices result in actual inverse using singular value decomposition, which is utilized by Matlab in computing the pseudoinverse [16].

In He *et al.* CRLB is numerically compared with the mean-squared error from maximum likelihood estimation, so that the difference of the mean squared error of the perfect estimator $\hat{\boldsymbol{\theta}}$ and CRLB

$$\left\langle \left(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\right)\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\right)^T \right\rangle_{\boldsymbol{r}|\boldsymbol{\theta},\boldsymbol{\alpha}} - \mathrm{CRLB}\left(\boldsymbol{\theta}|\boldsymbol{\alpha}\right) \tag{4.1}$$

is positive semidefinite [10]. Knowing that the diagonal elements of a Hermitian positive semidefinite matrix are real and non-negative, the scalar mean error per parameter in parameter vector $\boldsymbol{\theta}$ is derived simply by taking the square root of the appropriate diagonal element of the CRLB, which is *root-mean-squared error* (RMSE). This allows the comparison of any combination of parameters between simulation results.

4.1 Signal model

To assimilate relevance to real applications using public and common frequency modulated radio broadcasts, which operate in the $87.500 - 108.000$ MHz range in Finland [24], the carrier frequencies in theoretical simulations vary between 100 MHz and 100.4 MHz with 200 kHz intervals, using 75 kHz peak deviation, which

are based on actual implementation [5]. The baseband is sinusoidal wave of 7 kHz and increasing per illuminator with 100 Hz interval, chosen for simplified computing within a range, so that as a analogical wave in standard conditions the frequencies can be heard by human ear. A frequency modulated carrier [29, p.112] is represented mathematically as

$$s_c(t) = A\cos\left(2\pi f_c t + \phi(t)\right), \tag{4.2}$$

where $2\pi f_c t + \phi(t)$ is the instantaneous phase, and f_c is the carrier frequency. Since the baseband signal is also sinusoidal, the broadcasted wave is of the form

$$s_c(t) = A\cos\left(2\pi f_c t - h\cos(2\pi f_m t)\right), \tag{4.3}$$

where h is the modulation index $\Delta f / f_m$, f_m in the baseband frequency and Δf is the peak deviation.

Notably, in contrary to traditional radar implementation the relevant signal is continuous, instead of pulsating [23, p.1.5]. The cyclic form of the signal becomes an issue when the detection distance grows. In the case of 100 MHz carrier and 7 kHz baseband frequencies the waveform repeats every seventh baseband cycle, which gives the wave time to propagate almost 300 kilometres. Assuming the baseband was only 5 kHz, which is also in the range of audible sound, the waveform would repeat every baseband cycle, and the distance between indistinctive phases would be only fifth, some 60 kilometres. Given the only region of interest is the high SCNR region, and the formalized illuminators have some variance, this duplication of phase is assumed negligible.

The carrier frequency is the highest frequency component in the signal. Thus the signal is sampled at twice [29, p.78] the carrier frequency, 200Mhz. This may not be feasible in actual implementation, and one should therefore use the estimated lowpass equivalent of the time sampled signal, like proposed in [10]. The sampling rate of 200 MHz coincides with the sampling period $T_s = 5\times10^{-7}$ seconds. According to early test runs the amount of samples correlates strongly with computation time, and was limited to $k = 60$, resulting in one observation of a single object lasting 3×10^{-5} seconds. This seems unnecessarily and possibly unrealistically high resolution for any system operated or monitored by a person, but for purely computational exercise no obvious issues emerge, since actual implementation would use a low-pass

filtered version of the signal.

For electromagnetic radiation in the range of Very High Frequency (VHF) the higher atmospheric diffraction and other propagation effects differing from free-path propagation are ignored. Thus radio horizon may be denoted simply as a line-of-sight propagation range on the surface of Earth. Atmospheric refraction under normal conditions produces a wider range for VHF transmitters above ground level [26, p.33-13], but reflections and absorption from irregular terrain could diminish these to a great extend. The average height of Finnish FM VHF broadcasting antennae in 2016 was 114 meters above ground level [22]. Assuming a spherical Earth and a smooth surface, simple geometry would result in radio horizon beyond 38 kilometers for a transmitter at this height. An object 500 meters above ground level could be seen at almost 120 kilometers away, which is assumed to be beyond the high SNR region. Thus no terrain or atmospheric propagation effect are taken into effect. The model is however designed to be compatible with an arbitrary path based propagation model, naturally with increased computational cost.

4.2 Topological elements

In 2016 there were 1024 licensed radio broadcasters in Finland according to the Finnish Communications Regulatory Authority Viestintävirasto [22]. The effective radiated power of these stations spanned from 1.0 μW to 60 kW, with an average of 7.0 kW. There were 149 stations with effective radiated power of 10 kW or more, so 10 kW was decided to be a reasonable radiated power of a formalized illuminator.

The height, or z-coordinate in the simulation coordination of all illuminator and receiver antennae is 0 m to allow systematic comparison between topologies. Since no propagation effects of the terrain or atmosphere are taken into account, the absolute level of the components is irrelevant. All illuminators radiate isotropically.

The simulation computes a CRLB for evenly spaced set of coordinates. To have the most relevant data all target coordinates are located on a plane orthogonal to z-axis. Target object height was set to 500 meters to have some comparability to previous research [1,2], assuming antennae in those papers were somewhat over ground level.

4.3 Reflection coefficient

The reflected signal is never isotropic even from a spherical object, like assumed in the case of the topological elements, even though when spherical the reflection

coefficient is a constant. The coefficient denotes the ratio of the field strength of the reflected wave to that of the incident one. Since the reflective distribution is a property of the object, not of the system, and may depend on the position and orientation of the object, it is assumed to follow Gaussian distribution. The relative reflection coefficients in the matrix $\boldsymbol{R}$ are assumed spatially dependent as the function of the angle of signal to and from the object. In accordance to [10] the covariance matrix $\boldsymbol{R}$ is set as

$$\boldsymbol{R} = \boldsymbol{R}^r \otimes \boldsymbol{R}^i, \tag{4.4}$$

where

$$\boldsymbol{R}^r = \begin{bmatrix} \mu^r_{11} & \cdots & \mu^r_{1N} \\ \vdots & \ddots & \vdots \\ \mu^r_{N1} & \cdots & \mu^r_{NN} \end{bmatrix}, \tag{4.5}$$

$$\mu^r_{nn'} = \exp\left(-\varpi \Delta\phi^r_{nn'}\right), \tag{4.6}$$

$$\boldsymbol{R}^i = \begin{bmatrix} \mu^i_{11} & \cdots & \mu^i_{1M} \\ \vdots & \ddots & \vdots \\ \mu^i_{M1} & \cdots & \mu^i_{MM} \end{bmatrix}, \tag{4.7}$$

and

$$\mu^i_{mm'} = \exp\left(-\varpi \Delta\phi^i_{mm'}\right). \tag{4.8}$$

In the matrices the symbol ϖ is the exponential decay rate relative to the angle, which is chosen to be 0.01 using [10] as a reference. The $\Delta\phi^t_{nn'}$ and $\Delta\phi^r_{nn'}$ are the angles between between nth and n'th target-to-receiver paths, and mth and m'th target-to-illuminator paths, respectively. Notably, since the model only calculates the lower bound of the estimator and is aware of the object properties, no threshold value is defined for the object surface area.

4.4 Noise

The noise is simulated as spatially dependent, so that the noise level from one illuminator to another is a function of their relative distance. The covariance matrix

$\boldsymbol{Q}$ of Gaussian noise is generated in accordance to [10], and is of the form

$$\boldsymbol{Q} = \boldsymbol{\sigma}_w^2 \tilde{\boldsymbol{Q}} \otimes \boldsymbol{I}_K, \tag{4.9}$$

where $\boldsymbol{I}_K$ is $K \times K$ identity and the nn'th element is the exponential

$$\tilde{Q}_{nn'} = \exp\left(-d_{nn'}\gamma\right) \tag{4.10}$$

and γ is the parameter for exponential decay as a function of distance. Using [10] as a reference, γ is set to 5.0×10^{-5} and the variance matrix $\boldsymbol{\sigma}_w^2$ is 0.9 for all indices.

Since the intended receivers of FM broadcasts often have inefficient antennae and poor noise figures in addition to the source not being on line-of-sight, the transmitters are commonly considerably powerful [18, p.252]. This results in high noise levels like shown in the measurements in [18, p.254-255]. For instance, the countour plots in Figure 2.1 could be calculated with an extreme receiver effective noise figure of 50dB and isotropic antennae. This is because CRLB is not capable of accurately describing the estimators performance in low SCNR region [10], so the analysed areas lie within an area with high SCNR. According to [18] noise figures of 25dB are not pessimistic even in rural areas.

No specific elements of noise are isolated in this model, and the effective noise is estimated with free-space path loss. Implementing a propagation model taking surface and atmospheric effects into account as the noise model, it may be possible to achieve much more realistic results on the cost of computation time.

5. RESULTS

Some methods using partially or wholly precomputed data to predict the changes in coverage are investigated in this chapter. The scenarios use four illuminators of different baseband and carrier frequencies, but identical height and transmitted energy. The positions and identifiers of the illuminators are presented in Table 5.1. The raw simulated data covering some configurations of the illuminators are presented in Figure 5.1. Throughout this chapter the configurations in Figure 5.1 are referred to by the illuminator identifiers. For instance the single-illuminator configuration in the upper left part of aforementioned figure is referred to as the α-configuration, and the configuration on the upper right part of the same figure is referred to as the $\alpha\beta\gamma$-configuration. The compared 100 km $\times$100 km areas were simulated with a resolution of 500 $\times$ 500 evenly spaced data points, so one data point corresponds to $4 * 10^4$ m^2. Most of the simulations were computed on the Taito super cluster owned by CSC - IT Center for Science.

Identifier	x-coordinate	y-coordinate	z-coordinate
Alpha (α)	0	30000	0
Beta (β)	-25981	15000	0
Gamma (γ)	-25981	-15000	0
Delta (δ)	25981	-15000	0

Table 5.1 *Positions and identifiers of illuminators*

The simplest scenario with one illuminator is presented as the left uppermost diagram in Figure 5.1. The MSE consists of six estimated attributes: three spatial dimensions and three velocity components. Values of the mean squared error of different attributes, as well as the separate MSE of position and velocity are presented in Figures 5.3 and 5.2 respectively. It can be observed, that either the accuracy of velocity estimations, or the uncertainty of said estimations is great, since the lower bound of the variance is clearly lower for velocity, than position. Such opposite arguments for a single result further emphasize the applicability of the estimation algorithm to only exclude parts of the coverage area. Notably, the velocity com-

ponents contribute meticulously to the joint CRLB, and the joint MSE resembles closely the MSE of the position. The contribution of parameters of different scale to the MSE can also be detected in the plot in Figure 5.18. Notably, in the context of figures and text of this chapter most values are indicated as root-mean-squared error to illustrate better an estimation of the actual error.

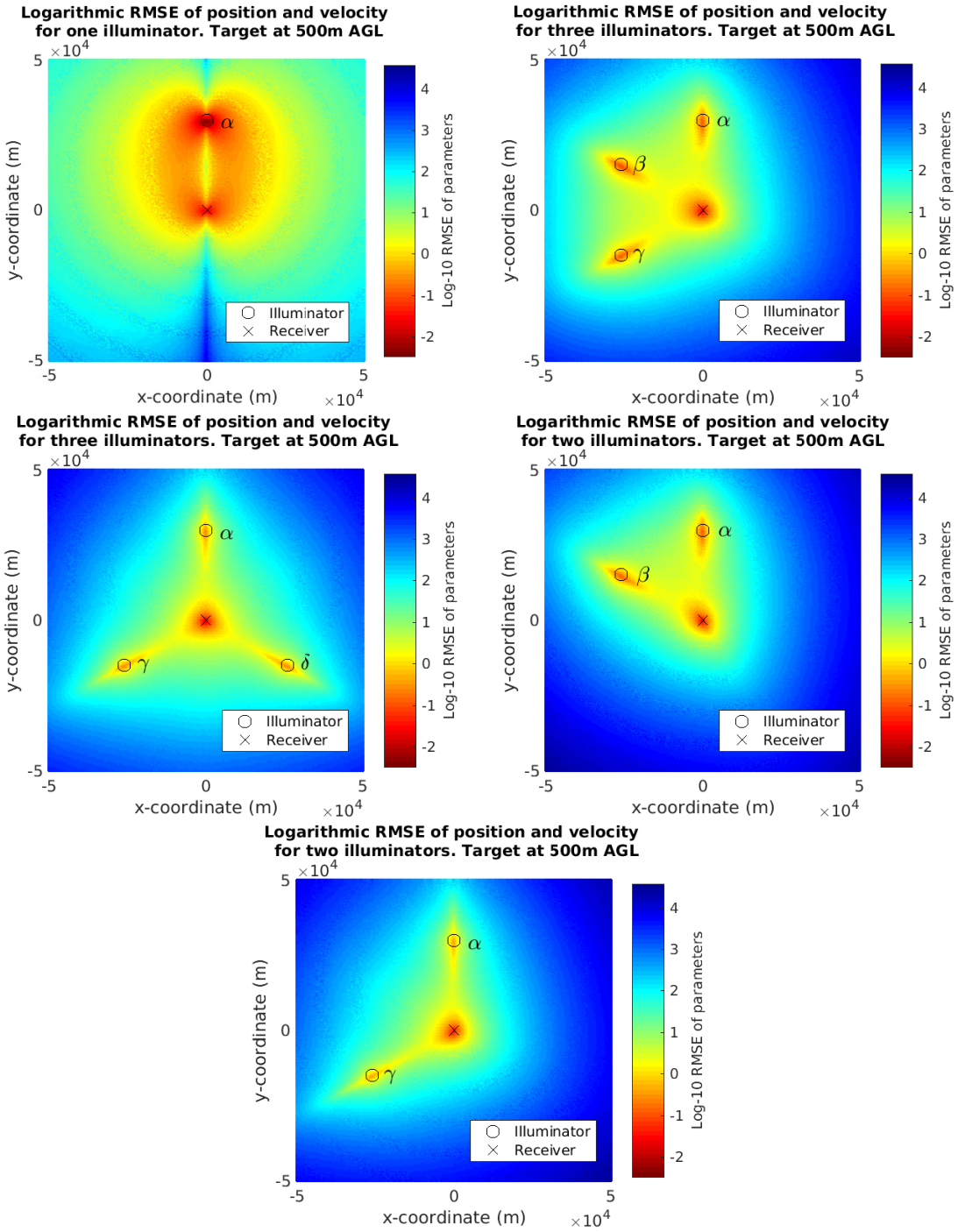

Figure 5.1 *Raw simulation results for different illuminator configurations with illuminator identifiers listed in Table 5.1*

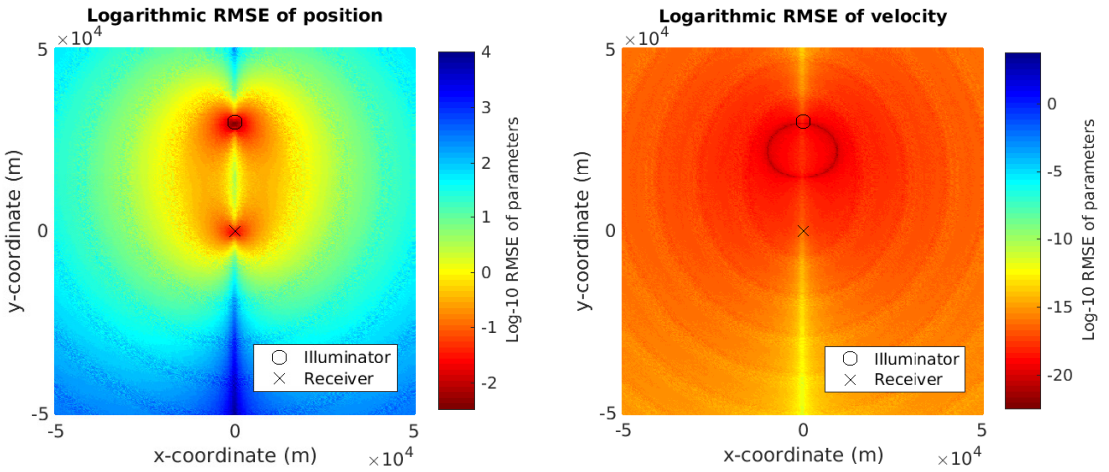

Figure 5.2 *On the left logarithmic RMSE for position only. On the right for velocity only. Target velocities are* $v_x = v_z = 0$ *and* $v_y = 0.5m/s$

It is apparent from the results in Figure 5.1 that there is a significant difference in the ability to limit the MSE between the single and multi-illuminator configurations when the distance from the z-plane set by the illuminator and the receiver increases. The area where the logarithmic RMSE is less than 0 for α-configuration in the upper left corner of Figure 5.1 is more than ten times what the area with the same criterion is for $\alpha\beta$-configuration in the middle right plot of the same figure, even thought multiple illuminators with different signal would presumably increase the accuracy. Since the values are the lower bound of the MSE, larger portion of desirably low logarithmic RMSE can be explained with the estimation algorithm having less information, and thus not being able to limit the MSE higher. This clearly indicates the applicability of the model for only decreasing the area in the coverage predictions. The difference is rather clearly observed in region around the line connecting the illuminator and the receiver in the single-illuminator configuration.

5.1 Varying parameters

Some of the constants involved with the model, such as the decay rates in the reflection coefficient and noise modeling were changed across orders of magnitude in various test cases, and some of effects were previously documented in [10]. In the test cases, no dramatic effect on the overall form of the MSE was observed, including doubling the relative height of the computed layer. The effects of modifying the illuminators to broadcast identical signals: consistent curves of distinct areas of higher and lower logarithmic RMSE bounds can be seen in Figure 5.4. Increasing the observing period scales with the computing time faster than linearly, which is expected based on the CRLB model. However, these are only computational

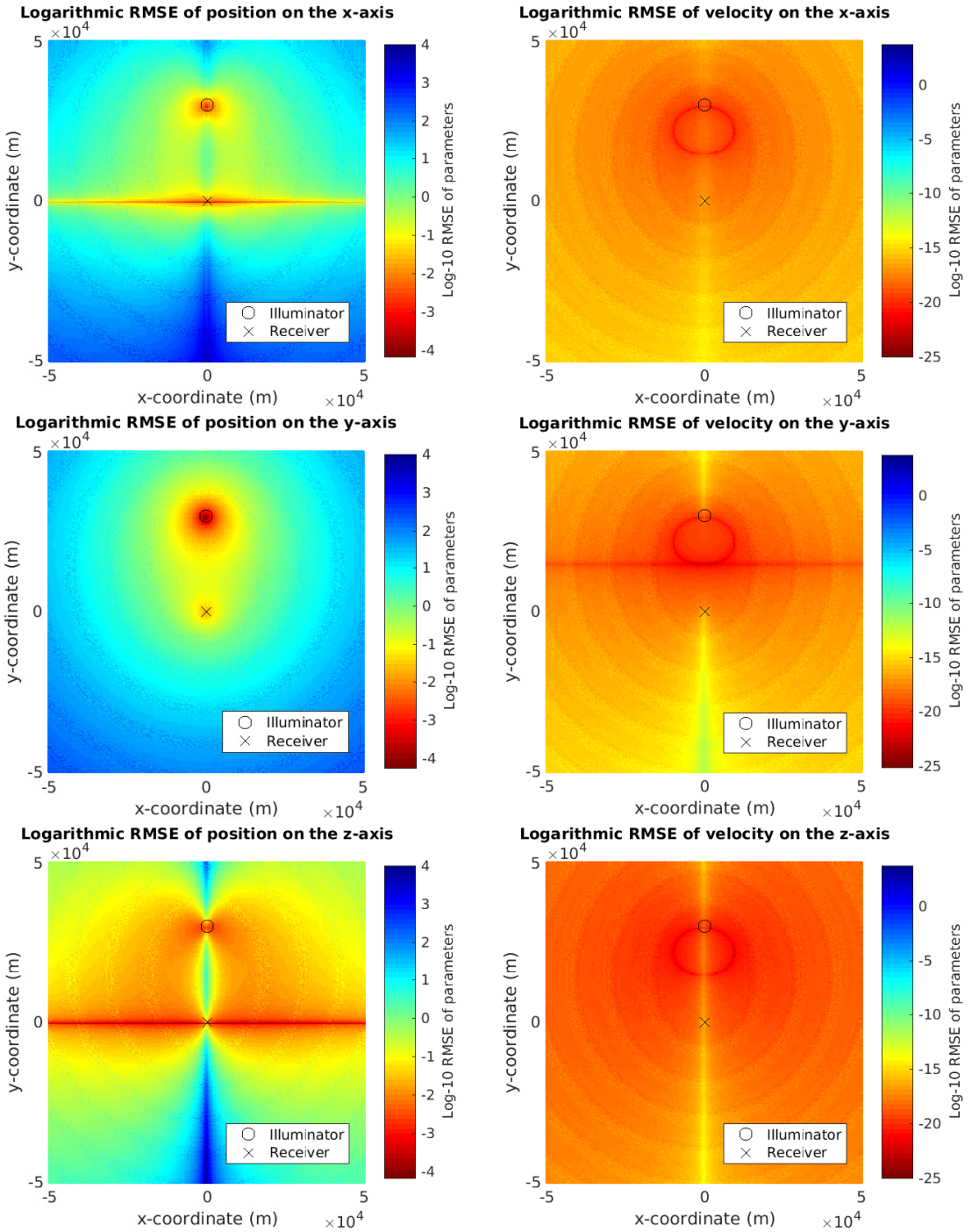

Figure 5.3 *Component-wise logarithmic RMSE of single-illuminator configuration. The different range of the colour scale for position and velocity components should be noted*

parameters, thus no decisive reference should be taken from the results in this thesis to actual measurements.

The lower right part of the Fisher information matrix containing the variances of the velocity components has values considerably lower than the position components. This can be observed in Figure 5.2, where the colour scales of position and velocity

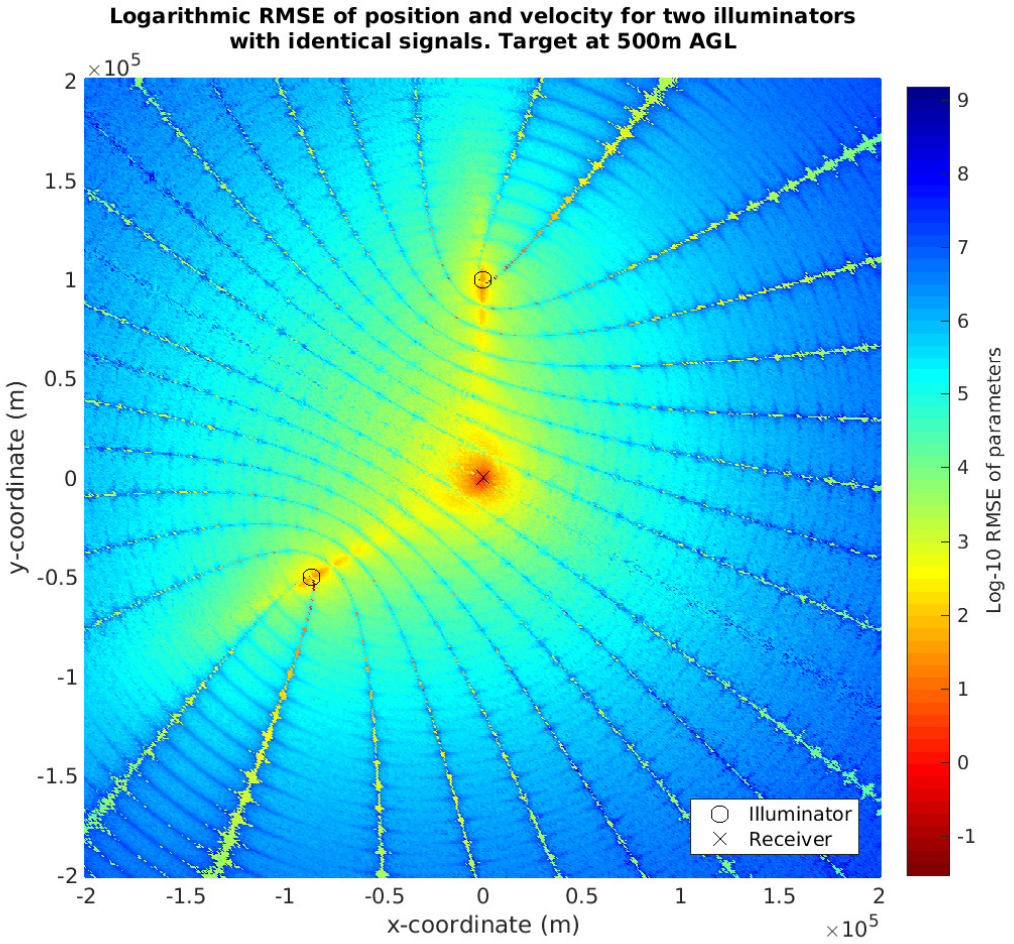

Figure 5.4 *Pattern of worse MSE lower bounds across the simulated area due multiple signal paths*

variances could not be aligned without loss of informativeness. The information about velocity is contained in the Doppler shift of the secondary signal, which is zero for all stationary objects. In addition, when stationary the covariance between the position components and the velocity components is zero.

The information matrix elements of velocities are in general between 10^{-10} and 10^{-30} even for all simulated values of the velocity vector, and the difference between the position and velocity RMSE in the same location are above ten orders of magnitude. Velocities from 0m/s to 100m/s were simulated distributed along different axes. For stationary targets the velocity block matrix reduces to zero when computing the pseudoinverse of a matrix that close to singularity. A zero-velocity vector would result in a uniformly distributed lower bound of zero due to the limitations in computation accuracy. Since the variance of stationary object velocity estimations cannot be visualised, and no relevant values of the velocity vector have any observable impact on the joint lower bound, the component-wise logarithmic RMSEs in

Figure 5.3 and 5.2 use a target velocity vector $\boldsymbol{v} = [0, 0.5, 0]^T$.

The waveform, which in this thesis is considered to be an external part of the model, is with decent certainty the most prone to error generation. Sampling the carrier frequency with extremely short sampling period, and an approximate normalization of the signal samples could results in almost 25% difference in the normalized signals samples between measurements. This can be verified by analysing the class *Illuminator* in Appendix B. In these simulations, where all points are assumed to be observed at the same instance in time, this proposes no issues with the relative results. Nonetheless, it is to be taken in consideration, if implemented as a time dependent model.

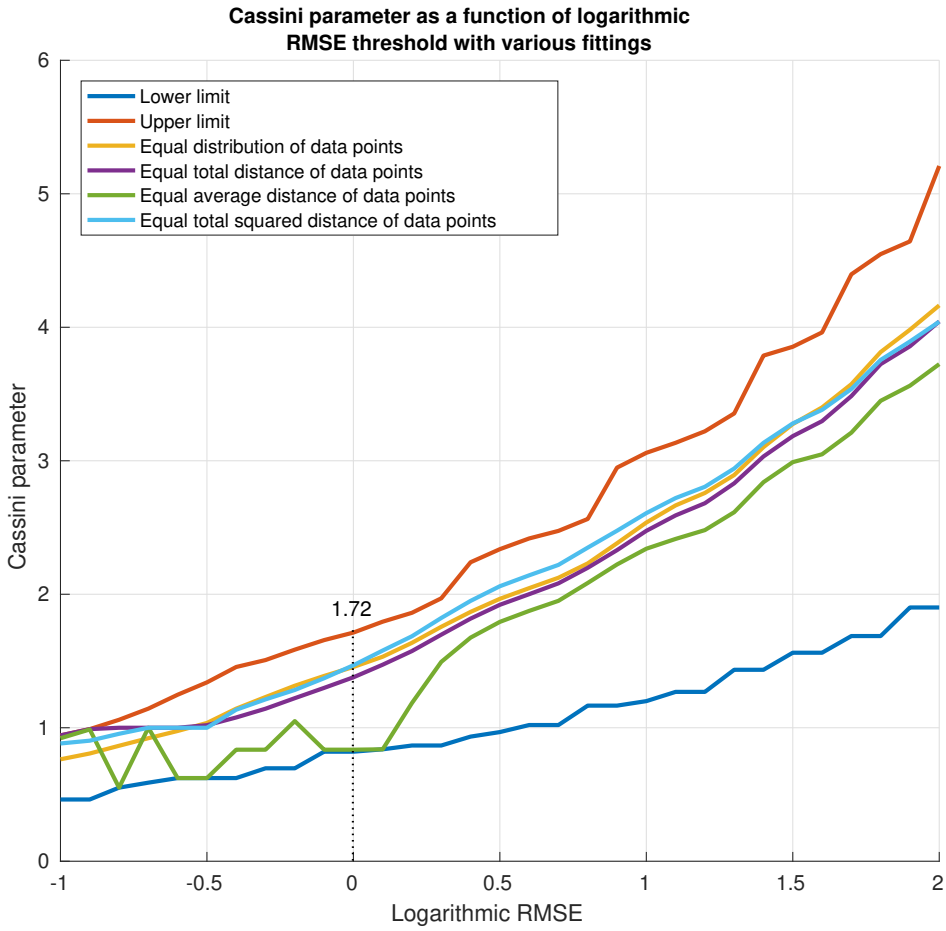

Figure 5.5 *Different fittings of Cassini parameter as the function of logarithmic RMSE. Irregular convergence is clearly visible with fitting the oval to average distance of data points for low logarithmic RMSE values*

5.2 Partial recomputing

In the event of topological change, one way to limit the recalculated area could theoretically be with the naïve approach of the illuminator, that has its status changed, does not affect area outside of the SCNR region dictated by the bistatic radar equation in Equation (2.1). The SCNR level limit defined as the Cassini oval depends on attributes of the system that are neglected in this model, such as antenna performance. Therefore the Cassini parameter can be chosen in many different ways based on the data. For actual feasibility of such decisions one would need to have knowledge of the illuminators' setup and the signal wave form.

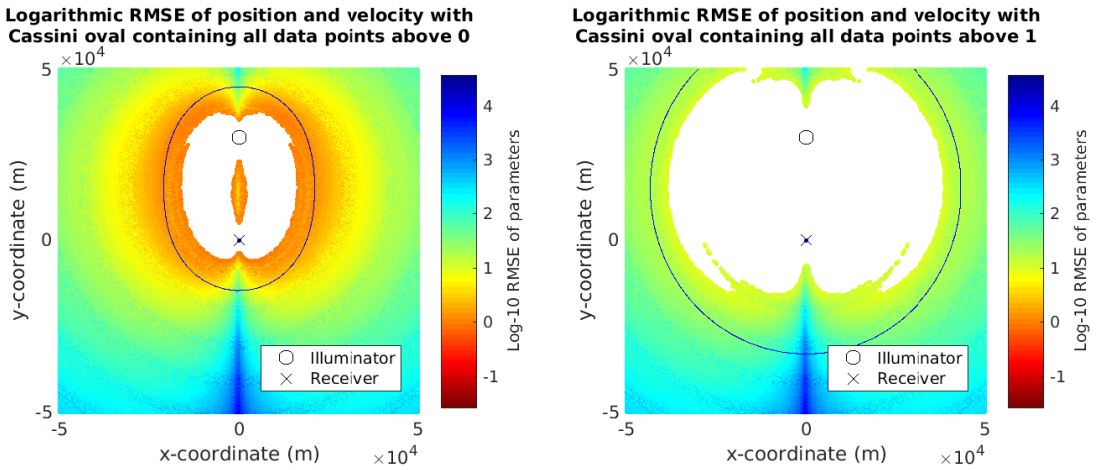

Figure 5.6 *Upper bound Cassini ovals for RMSE of 0 and 1 with a single illuminator*

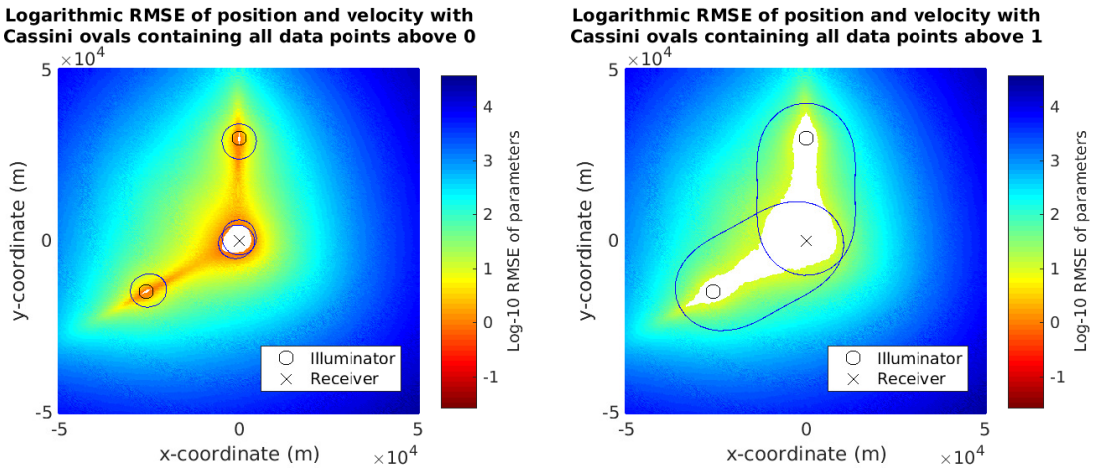

Figure 5.7 *Upper bound Cassini ovals for RMSE of 0 and 1 with multiple illuminator*

For a chosen RMSE threshold an oval of some initial Cassini parameter value was generated as a contour plot of appropriate z-level using the *fcountour* [16] functionality of MATLAB. The *inpolygon* [16] function was utilized to generate two sets of data points: the points with RMSE below the threshold outside the oval, and the

points with RMSE above the threshold inside the oval. Then, using the bisection method the value of the parameter that equalizes some attribute of the sets was solved. The attributes, based on which the ovals were fitted were number of data points and the distance, squared distance and average distance of the points from the oval. The different relations of the Cassini parameter, and thus the area to be recomputed, to the RMSE threshold are presented in Figure 5.5. The values are for a single illuminator, but two values for multiple illuminators are presented in Table 5.2. The upper bound is the value, at which there is no data below the threshold outside of the oval. Similarly the lower bound is the value, at which there is no data above the threshold inside the oval. Both of these have clear steps indicating the regularly appearing zones of increased noise, which can be visually verified from component-wise data in Figure 5.3. The increase in variance of the RMSE results in an increasing difference in the absolute values of upper and lower bounds of the Cassini parameter as the RMSE grows. The equal distribution of points of interest on both, inside and outside of the oval yields a smoother curve than the limits. In addition the values are similar whether the equal distribution is based on the number of points, the total distance of points from oval, or the total distance squared. For values around and below 0 the average distance tends to converge to the limits and fails to increase monotonically, which is not surprising considering the non-monotonic nature of average comparison, when the summands are transferred between the sets.

	Threshold	Cassini parameter
Single	0	1.72
	1	3.06
Multi	0	0.8
	1	1.35

Table 5.2 *The upper bound Cassini parameter values in Figures 5.6 and 5.7 for different logarithmic RMSE threshold values separately for single and multiple-illuminator cases*

Since the aim is to resolve whether similar coverage predictions can be achieved by partial recomputing, the upper bound Cassini parameter is chosen as the are limiting parameter. The highest parameter value maximizes the recomputed area, and thus is most likely to produce desirable results. If the difference in area is considerable even with the upper bound Cassini parameter of the prior area, it is not reasonable to assume better results would be achieved with lower parameter values.

Recomputing an area of simulation limited by strict bound results in discontinuous parts, so they may not be used to evaluate the performance of the estimator outside of chosen threshold. However, the interest in these combinations is solely the total

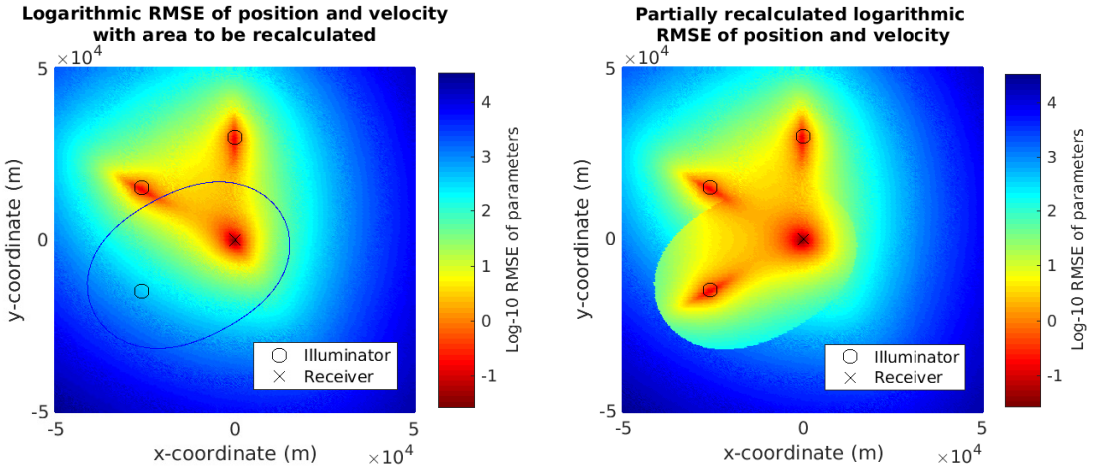

Figure 5.8 *The process of computing the limit and recalculating the limited area in transition from $\alpha\beta$-configuration to $\alpha\beta\gamma$-configuration. Cassini parameter is* 1.72

area limited by the RMSE threshold value, and the recomputed areas can, to some extend, be used to evaluate the accuracy of partly recalculated estimates.

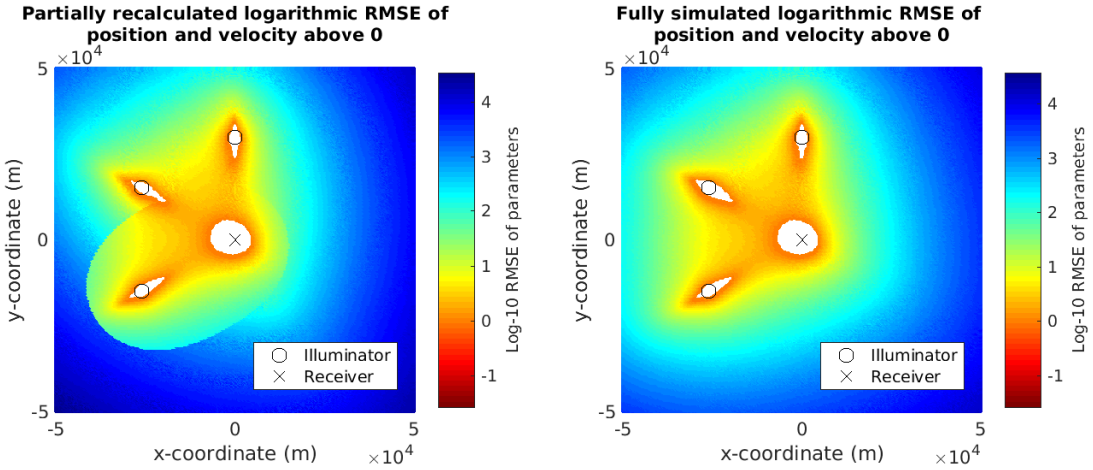

Figure 5.9 *Comparison between partly recalculated area and fully simulated area, when the logarithmic RMSE < 0, Cassini parameter is* 1.72*, and γ-illuminator is added to $\alpha\beta$-configuration*

Multiple-illuminator configurations have significantly higher bounds in comparison to the single-illuminator configuration. When multiple, the positioning of the illuminators has little effect on the lower bound of Cassini parameter for chosen RMSE limit. Therefore two values for the parameter were iteratively chosen for most investigated RMSE threshold values: one for a single illuminator visualised in Figure 5.6, and one for multiple-illuminator configurations, visualised in Figure 5.7. The values of the parameters are presented in Table 5.2.

Case	Threshold	Combination area (Data points)	Simulation area (Data points)	difference(%)
$\alpha\gamma \rightarrow \alpha\beta\gamma$	0	4510	5753	-21.6
$\alpha\beta \rightarrow \alpha\beta\gamma$	0	5432	5753	-5.6
$\alpha\beta\gamma \rightarrow \alpha\gamma$	0	3316	2073	60.0
$\alpha\beta\gamma \rightarrow \alpha\beta$	0	3733	2073	80.1
$\alpha \rightarrow \alpha\beta$(Prec.)	0	52724	3412	1445.3
$\alpha \rightarrow \alpha\beta$	0	15777	3412	362.4
CMSE(α, β)	0	16442	3412	381.9
CMSE(α, β, γ)	0	6732	5753	17.0
MIN(α, β)	0	56697	3412	156.2
MIN$(\alpha\beta, \gamma)$	0	41250	5753	617.0
MIN$(\alpha\gamma, \alpha\beta)$	0	3886	5753	-32.5
$\alpha\gamma \rightarrow \alpha\beta\gamma$	1	35417	46209	-23.4
$\alpha\beta \rightarrow \alpha\beta\gamma$	1	42861	46209	-7.2
$\alpha\beta\gamma \rightarrow \alpha\gamma$	1	30516	19778	54.3
$\alpha\beta\gamma \rightarrow \alpha\beta$	1	32621	29273	11.4
MIN$(\alpha\gamma, \alpha\beta)$	1	35164	46209	23.9

Table 5.3 *Areas above threshold in with different changes in topology. CMSE means combined mean squared error. MIN means minimum taken in each point with respect to the operands. (Prec.) implies the areas are precomputed as an elementary result. All combined MSE and minimum combinations are generated with precomputed results.*

The progress from limiting the area to recomputing the MSE is presented in Figure 5.8. Two partially recomputed results are presented in Figures 5.9 and 5.10. A total of eight such results were calculated: two additions of illuminators β and γ, and two removals of the same elements. All these were done with two RMSE thresholds: 0 and 1. The rest of the graphical results are found in Appendix C: Graphical results of partial recomputing. The difference in data points below the threshold is presented in Table 5.3, where the notation $\alpha \rightarrow \alpha\beta$ means the transition from one-illuminator α-configuration to two-illuminator $\alpha\beta$-configuration. In other words the addition of β-illuminator. Similarly $\alpha\beta \rightarrow \alpha$ would mean the removal on β-illuminator. Cases for threshold value 1 are provided mainly for visual reference of the Cassini oval apparently bounding the threshold area. Recomputed areas, where the threshold is set to 0, are calculated using the large Cassini parameter bounding also the single-illuminator configuration. When the threshold is indicated to be 1, the Cassini parameter used to bound the recomputed area is the one of multi-illuminator case, as it provides effectively a tighter bound for the area being

recomputed, and using the single-illuminator value would give incomparable results since the recomputed areas would exceed the simulated area.

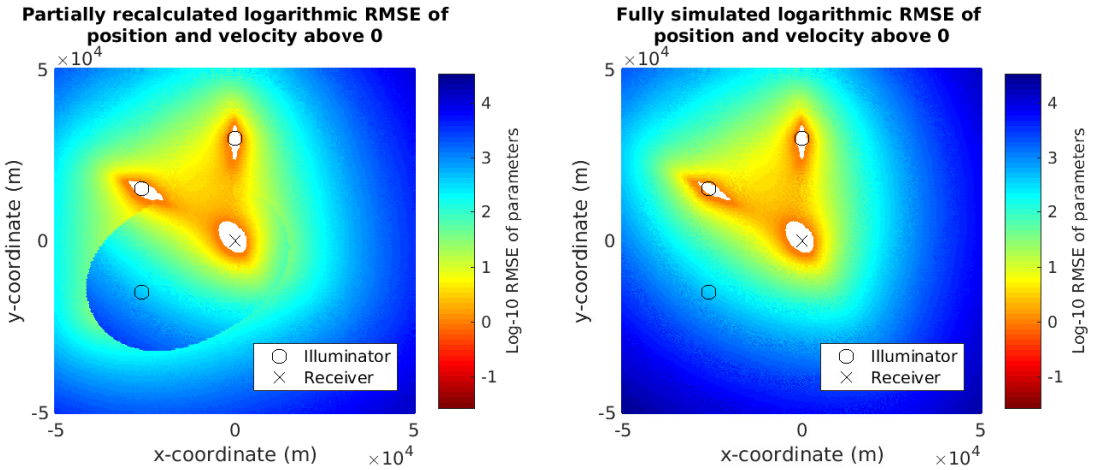

Figure 5.10 *Comparison between partly recalculated area and fully simulated area, when the logarithmic* $RMSE < 0$*, Cassini parameter is* 1.72*, and* γ*-illuminator is removed from* $\alpha\beta\gamma$*-configuration*

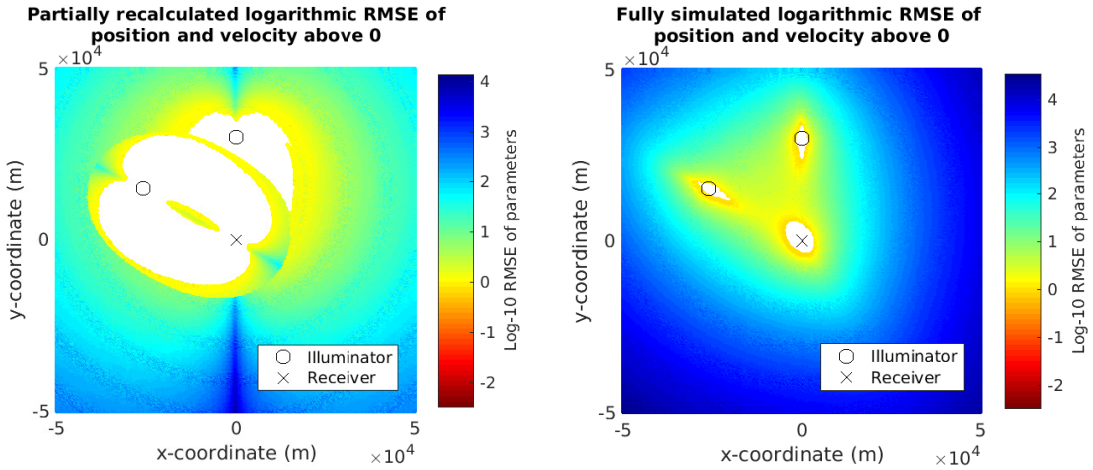

Figure 5.11 *Comparison between fully but separately simulated area and fully simulated area, when* $MSE < 0$*, Cassini parameter is* 1.72*, and precomputed* β*-illuminator is added to* α*-configuration*

To avoid some degree of discontinuity seemingly atypical for the MSE, the recalculation of area should be limited to multiple-illuminator configurations. The results in Figures 5.11 and 5.12 visualise the problems in the continuity of the area above the MSE threshold emerging from the very low limits of single-illuminator configuration. From Table 5.3 it could be speculated, that the type of change, as in adding or removing illuminators, and the system configuration have the largest effect on the areal difference. For instance, the difference in change $\alpha\beta \rightarrow \alpha\beta\gamma$ had a relative difference of less than half of any other change regardless of the threshold.

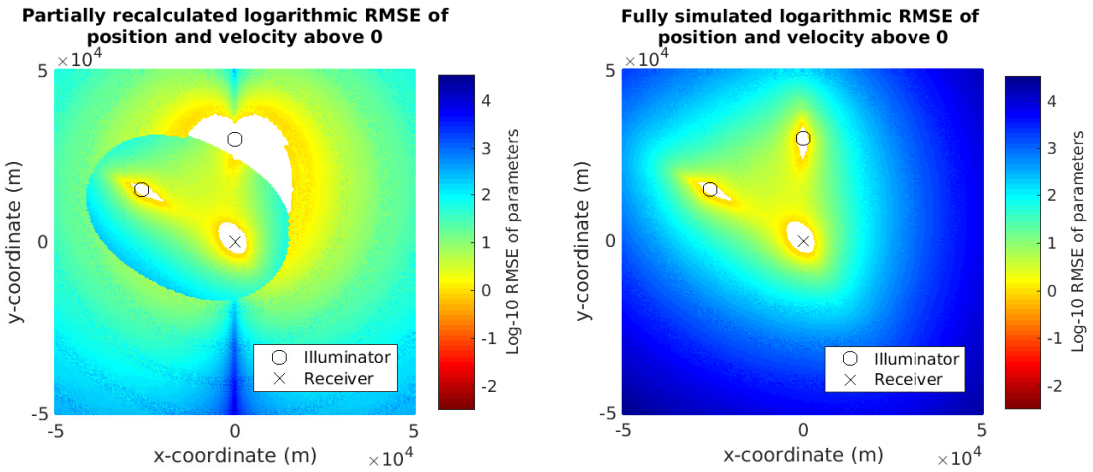

Figure 5.12 *Comparison between partly recalculated area and fully simulated area, when* $MSE < 0$*, Cassini parameter is* 1.72*, and* β*-illuminator is added to* α*-configuration*

With free-path propagation all results indicate a pleasant smoothness across the simulated space. This makes the CRLB results easily interpolated. Notably, implementing a sophisticated propagating model yields much higher levels of gradient especially closer to ground level. This is clearly indicated in [1, 4]. Two cases with sparsely calculated limited areas are presented in Figures 5.13 and 5.14. Comparison to the corresponding figures of the full simulation in Appendix C shows clear resemblance. The interpolated areas are calculated with 1/5th resolution. That is less than 4% of the reference points, which are exhaustive to compute. Even with the larger of the used ovals, the computation time of partial recalculation is less than 4% of the whole simulation. Here it should be reminded, that calculations with increasing amount of illuminators are slower. This means, that a consumer level modern desktop computer can manage the partial recalculations, including the interpolation in a matter of minutes, compared to around 2 hours for the full simulation.

5.3 Precomputed combination

Computing the most elementary single-illuminator cases and combining them either linearly or otherwise would greatly enhance the modularity of the computations, since the memory and processing requirements increase as the amount of topological elements in the simulation increases. Using precomputed illuminator-receiver pairs yields apparently invalid results if the parameter-wise MSE are combined. Combined MSE of two and three single pairs are presented in Figures 5.15 and 5.16. The uncertainty of a single-illuminator configuration gives harshly lower limits, and the areas on the line through the topological elements suffer greatly from, what can be

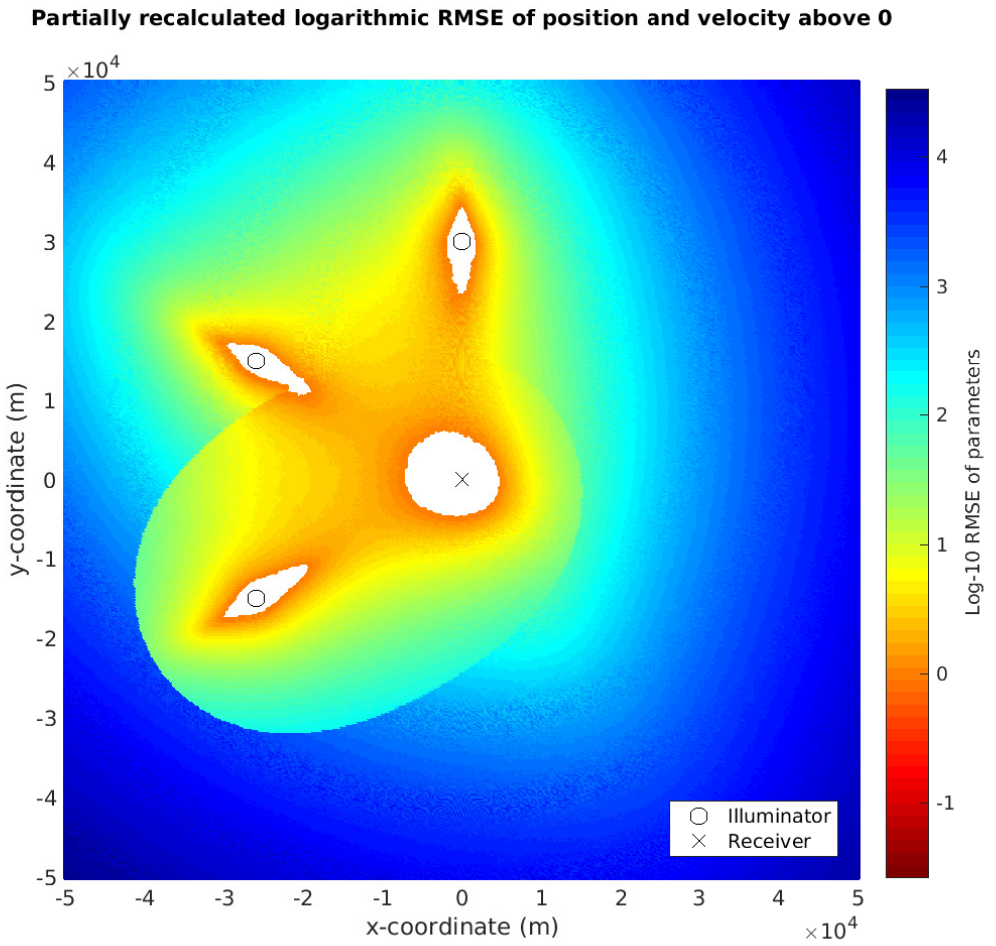

Figure 5.13 *Partially recomputed logarithmic RMSE with full resolution of* 500×500 *data points when RMSE* > 0

assumed to be noise from the direct signal, since that is the single greatest element of the noise.

In Figures 5.15 and 5.16 the combined RMSE of single-illuminator cases indicate the unsuitability of linearly combining elementary results, when compared to the calculated results of the same topologies in Figure 5.1. With multiple illuminators however, the results seem more appropriate. Combined RMSE and minimum RMSE taken in regard to precomputed two-lluminator cases are shown in Figure 5.18, with values along the line $x = -10^4$ plotted in Figure 5.18. A more familiar form is observed comparing it to the full simulation results in Figure 5.1, but only with the selective RMSE. With combined RMSE clear interference from overlapping illuminators diminishes the rest of the desired region, as expected from the way MSE is constructed.

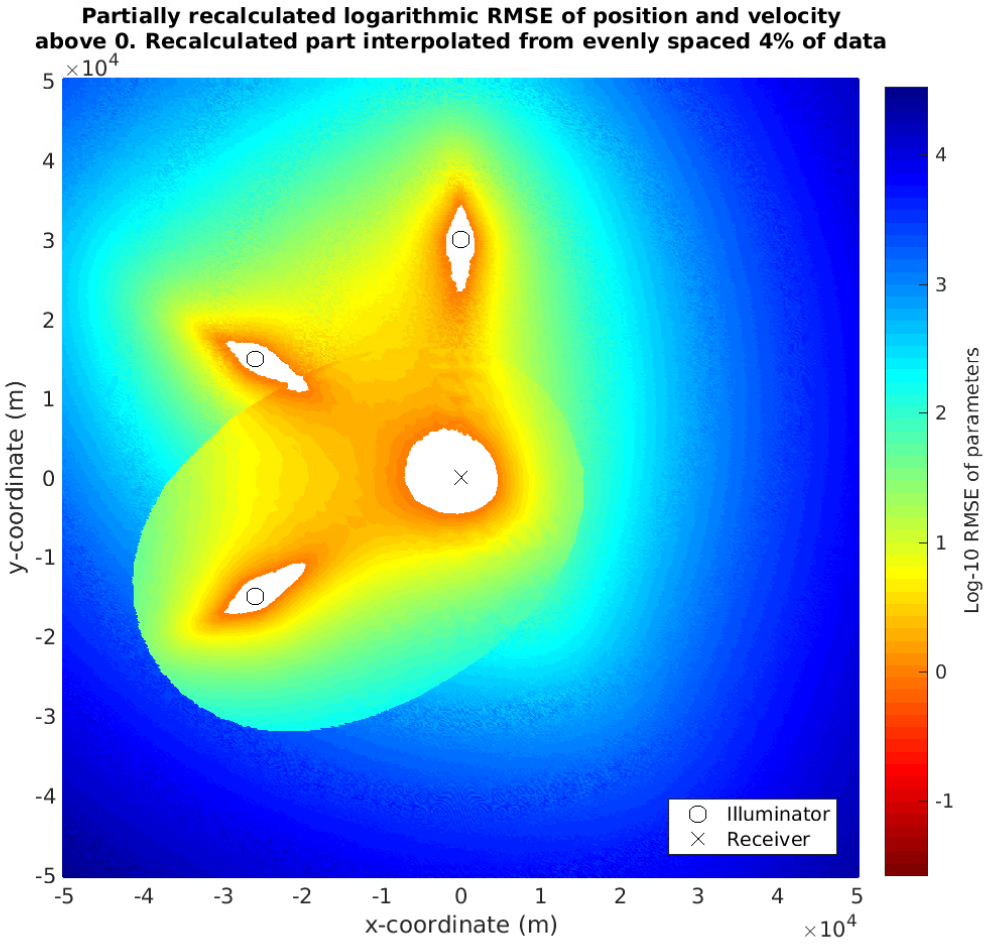

Figure 5.14 *Partially recomputed logarithmic RMSE when RMSE > 1. The recomputed area is interpolated from data points with* $1/5$*th resolution.*

5.4 Elementary coverage volume

The model presented in this thesis is a three-dimensional extension of the original model by He *et al.* [10], and capable of computing the MSE of an arbitrary point. All volume predictions are absent due to the computational limitations set even for one layer with the used resolution. However, single-point calculations can be used to verify the analogy between two and three-dimensional results given ideally isotropic antennae. From observations throughout this chapter it is expected, that with certain assumptions the logarithmic RMSE volume could be efficiently computed for a single-illuminator configuration.

The contributions of the velocity diminish in the mean squared error due to the much more significant contributions of the location. Setting the apparently insignificant velocity vector close to zero the MSE of one illuminator-receiver pair with isotropic elements can be approximated rotationally symmetric around an arbitrary

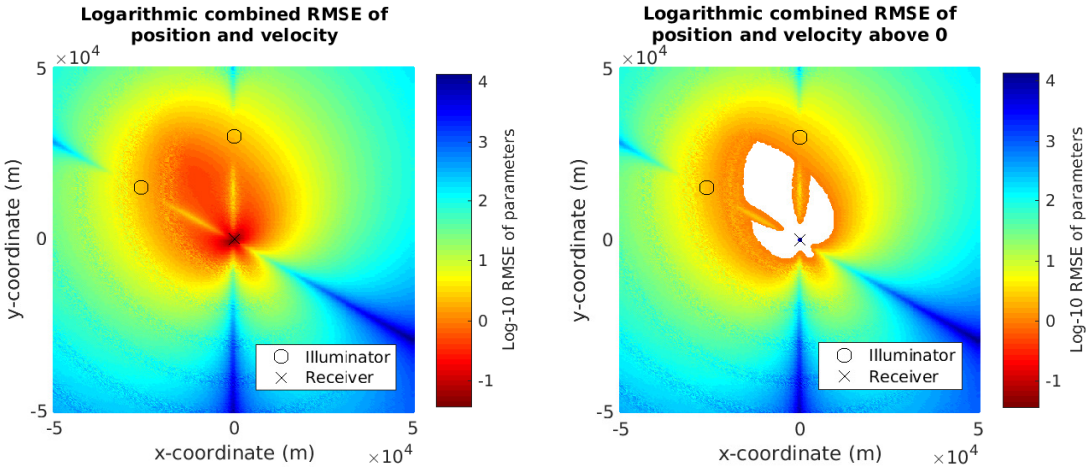

Figure 5.15 *Logarithmic combined RMSE of two single pairs and threshold area, when* $MSE < 0$

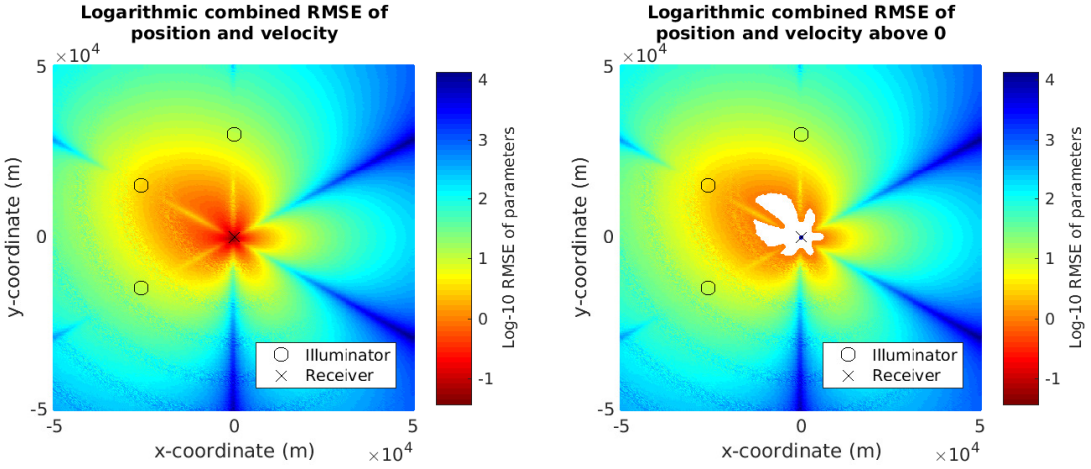

Figure 5.16 *Logarithmic combined RMSE of three single pairs and threshold area, when* $MSE < 0$

illuminator-receiver line. Knowing this, a cylindrical volume can be computed with half of the computations of one layer, such as the layer presented in Figure 5.1. With the symmetry in combination with the interpolated lower-resolution data the cylindrical volume of one pair could be calculated in time many orders of magnitude shorter, than a full simulation. Using the 1/5th resolution from Subsection 5.2 the number of data points needed is $1/(250 * \pi) \approx 0.13\%$ of the number needed to fully simulate an identical volume. However, the results in Figure 5.15 and 5.16 indicate that no method to combine the precalculated single-illuminator configurations has yielded acceptable outcome. This is discouraging since the improvement in volumetric predictions is dependent on finding a valid method to combine elementary mean squared errors.

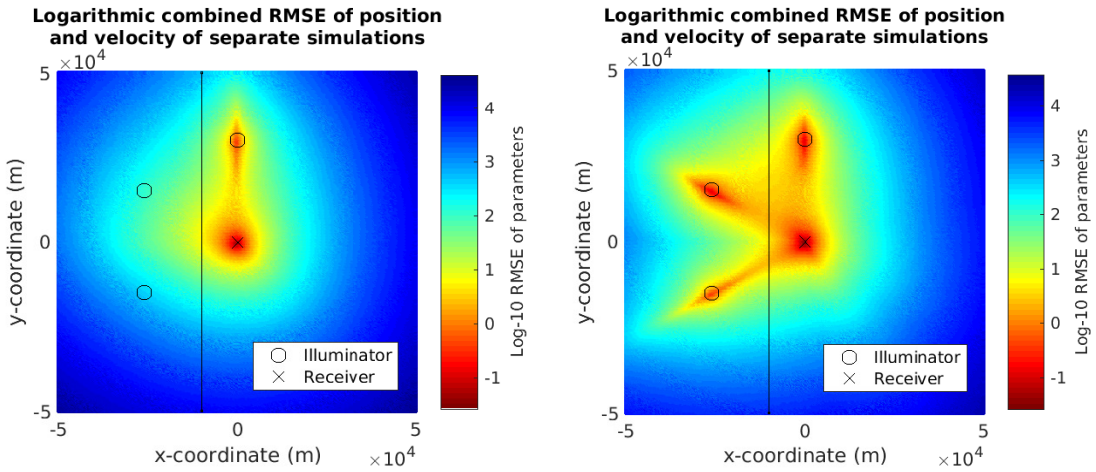

Figure 5.17 *Logarithmic combined RMSE and minimum presentation of joined $\alpha\beta$ and $\alpha\gamma$ configurations. Line at $x = -10^4$ represents the plotted MSE*

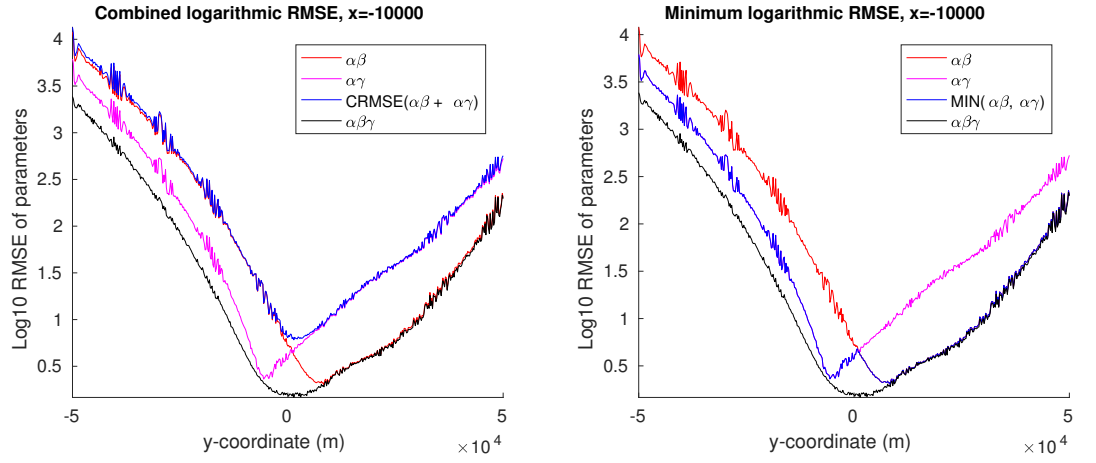

Figure 5.18 *Logarithmic RMSE along the line $x = -10^4$. Blue line is the combined MSE on the left, and MIN on the right. The poor ability and form of simple combined estimations and selective estimations can be seen in the difference of the blue and black lines*

5.5 Simulation with an existing tolopogy

The simulation result in Figure 5.19 is provided as a curiosity and vague reference point of the public FM-radio broadcasters near Jyväskylä, Finland. In the same figure are also the areas limited by logarithmic $RMSE > 2$, $RMSE > 1$ and $RMSE > 0$. The antenna height, effective radiated power, carrier frequency and coordinates where parsed from the public data of the Finnish Communications Regulatory Authority Viestintävirasto [22], and the ground elevations were based on Google Maps data, accessed via online tool [6]. Most of the broadcast locations had more than one station, but only one signal per location is monitored. The stations were chosen by their effective radiated power and frequency separation to other stations. The receiver is 259m above sea level and the antenna is located on

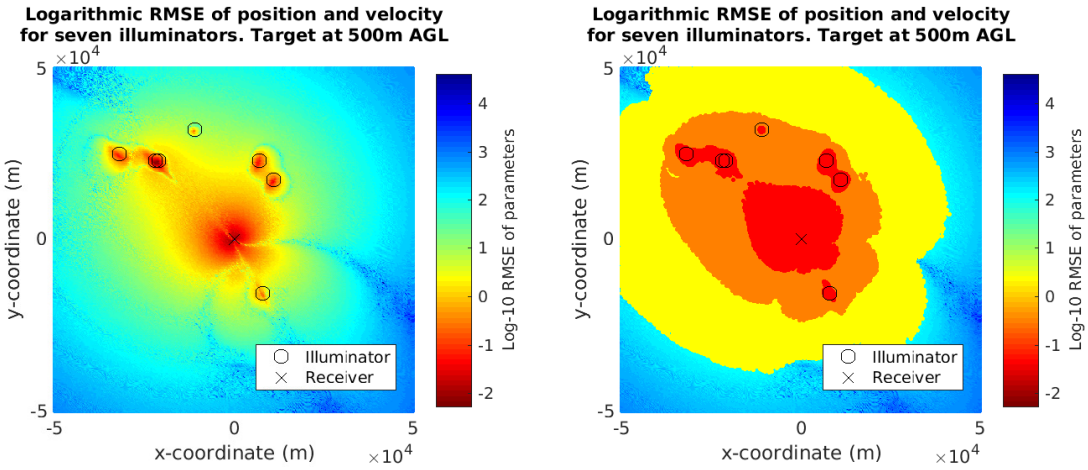

Figure 5.19 *Logarithmic RMSE of seven actual FM-radio broadcasters at differing heights. In the right hand side picture the uniform areas of red, orange and yellow do not follow the colour map on the right. Yellow region represents all* $RMSE < 2$*, orange* $RMSE < 1$ *and red* $RMSE < 0$

top of a 12m vertical mast.

6. CONCLUSIONS

This thesis has explored some designs to modularize a large computation of passive radar Cramér-Rao lower bound. Utilizing a simulation engine to compute CRLB over a plane, the applicability of different methods to partition and combine smaller results were investigated. Computation time was alleviated with numerous assumptions of ideal behaviour, thus yielding results relative only to the chosen environment. One investigated approach was to limit the area requiring recalculation in case of topological change, with some, not necessarily right for the corresponding antenna configuration, solution to bistatic radar equation. Others were calculating the combined MSE of multiple simulations, and using different simulation results selectively.

The situations where recalculating an area limited with by some Cassini parameter had promising results were uncommon. The inability of a single illuminator to limit its vicinity to much extend means partial recalculation could only be performed, when neither the original nor resulted configuration were single-illuminator. In addition, most results with higher than zero logarithmic root-mean-squared error threshold value were of much worse quality than those of zero threshold. Even when it is stated by He *et al.* [10], that "the Cramér-Rao lower bound is a tight bound only for the high SCNR region" and is "incapable of accurately describing the low-SCNR estimation performance of estimators", the results would indicate unacceptable losses in accuracy. From results in Figures 4, 5, 6 and 7 in Appendix C it is evident, that the upper bound Cassini parameter set even for a multi-illuminator configuration for some threshold error does not actually bound the MSE threshold area induced by the changing illuminator. Although it can not be shown from results in this thesis, the author suspects using some other estimate of the Cassini parameter as the bound for error threshold area achieves at most miniscule benefits compared to a full simulation.

The simulations did not show, apart from partly recomputed configurations, especially large gradients , which could be expected based on previous work. This can probably be credited to the lack of extensive propagation model. However, the interpolated results were comparable to the full resolution simulation with only a

few percent of computation required. Partial calculation could therefore be used as a rapid adjustment model, prioritizing vastly interpolated areas within lower Cassini parameter, and improving resolution when deemed necessary. If one is only interested in the threshold of some MSE, it could be achieved swiftly focusing recomputing in the area near the threshold, although like stated before, the CRLB model does not perform well in low-SCNR region.

Combining the elementary data to one mean squared error, or selectively choosing the data to use achieves undesirable results, which can be analytically verified with generalized CRLB for joint estimation of position and velocity [10]. However the set of more elementary results with one pair combinations versus all combinations exhibits linear growth versus exponential, assuming one receiver. Thus even combining the elementary results in a complex way could be profitable as the amount of illuminators grows. To achieve such method, analysis on the CRLB algorithm is required.

In the end the generalized Cramér-Rao lower bound did perform on the expected level. Its most imperative shortcoming, the inability to describe the low-SCNR region, was well analysed by He *et al.* [10], and could therefore be accredited when analysing the data. The model used is hardly an efficient method to calculate coverage areas, or space even less. Some improvements can be achieved with extensive interpolation, since the estimation algorithm performs well in single point examination.

There is room for further research, but actual and verified measurement data is favored in order to determine the direction of research focus. Especially results applying a refined propagation model to the signal and noise models could be compared to the previous coverage research. For actual performance comparison with an estimation algorithm the codes presented in Appendix B could easily be extended to cover the mismatched case of some parameters, which is thoroughly explained in [10].

Although no conclusive outcome was achieved regarding different ways to partition the calculation, it is the belief of the author, that by analysing the CRLB in more detail a method could be derived to achieve reasonable modularity at least with a linear propagation model.

BIBLIOGRAPHY

[1] W. C. Barott, T. Dabrowski, and B. Himed, "Fidelity and complexity in passive radar simulations," in *2015 IEEE 16th International Symposium on High Assurance Systems Engineering*, Jan 2015, pp. 277–278.

[2] W. C. Barott and B. Himed, "Simulation model for wide-area multi-service passive radar coverage predictions," in *2013 IEEE Radar Conference (RadarCon13)*, April 2013, pp. 1–4.

[3] A. v. d. Bos, *Parameter Estimation for Scientists and Engineers*. Wiley-Interscience, 2007.

[4] B. Chan, "Receiver site optimisation for passive coherent location (pcl) radar system," Master's thesis, University of Cape Town, 2008. [Online]. Available: http://rrsg.ee.uct.ac.za/theses/ug_projects/chan_ugthesis.pdf, Accessed 20.9.2016

[5] L. Der. (2015) Frequency modulation (fm) tutorial. [Online]. Available: http://www.silabs.com/Marcom%20Documents/Resources/FMTutorial.pdf, Accessed 25.11.2016

[6] Elevation Finder. Freemaptools. 2016. [Online]. Available: https://www.freemaptools.com/elevation-finder.htm, Accessed 25.11.2016

[7] B. Frieden, *Science from Fisher information : a unification*. Cambridge, UK New York: Cambridge University Press, 2004.

[8] R. G. Gallager, "Circularly-symmetric gaussian random vectors," *preprint*, pp. 1–9, 2008.

[9] N. R. Goodman, "Statistical analysis based on a certain multivariate complex gaussian distribution (an introduction)," *The Annals of Mathematical Statistics*, vol. 34, no. 1, pp. 152–177, 1963.

[10] Q. He, J. Hu, R. S. Blum, and Y. Wu, "Generalized cramér-rao bound for joint estimation of target position and velocity for active and passive radar networks," *IEEE Transactions on Signal Processing*, vol. 64, no. 8, pp. 2078–2089, April 2016.

[11] P. Hoff, *A First Course in Bayesian Statistical Methods*, ser. Springer Texts in Statistics. Springer New York, 2009.

[12] G. Hufford. (1995) The its irregular terrain model. [Online]. Available: http://www.its.bldrdoc.gov/media/50676/itm_alg.pdf, Accessed 28.11.2016

[13] W. Jun, Z. Shouhong, and B. Zheng, "Passive coherent detection via exploiting illuminator of opportunity," in *6th International Conference on Signal Processing, 2002.*, vol. 2, Aug 2002, pp. 1415–1418 vol.2.

[14] S. Kay, *Fundamentals of Statistical Signal Processing: Detection theory*, ser. Fundamentals of Statistical Signal Processing. PTR Prentice-Hall, 1998.

[15] E. Lehmann and G. Casella, *Theory of Point Estimation*, ser. Springer Texts in Statistics. Springer New York, 2003.

[16] MATLAB version 9.1 (r2016b) online documentation. The MathWorks Inc. Natick, Massachusetts. [Online]. Available: https://se.mathworks.com/help/matlab/, Accessed 17.1.2017

[17] K. Mardia, J. Kent, and J. Bibby, *Multivariate analysis*, ser. Probability and mathematical statistics. Academic Press, 1979.

[18] A. Moccia, M. D'Errico, A. Moreira, G. Krieger, P. Dubois-Fernandez, H. Cantalloube, B. Vaizan, M. Cherniakov, T. Zeng, P. Howland, H. Griffiths, C. Baker, and J. Sahr, *Bistatic Radar: Emerging Technology*, M. Cherniakov, Ed. "John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester,West Sussex PO19 8SQ, England", 2008.

[19] W. L. Patterson, "Advanced refractive effects prediction system (areps)," in *2007 IEEE Radar Conference*, April 2007, pp. 891–895.

[20] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," nov 2012, version 20121115. [Online]. Available: http://www2.imm.dtu.dk/pubdb/p.php?3274

[21] P. Ptak, J. Hartikka, M. Ritola, and T. Kauranne, "Long-distance multistatic aircraft tracking with vhf frequency doppler effect," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 3, pp. 2242–2252, July 2014.

[22] Radio stations in finland. Viestintävirasto. 2016. [Online]. Available: https://www.viestintavirasto.fi/en/spectrum/radiospectrumuse/radiostationsinfinland.html, Accessed 25.11.2016

[23] M. Skolnik, *Radar Handbook, Third Edition*, ser. Electronics electrical engineering. McGraw-Hill Education, 2008.

[24] Taajuusjakotaulukko. Viestintävirasto. 2015. [Online]. Available: https://www.viestintavirasto.fi/attachments/maaraykset/Taajuusjakotaulukko_M4T_15.12.2015.pdf, Accessed 25.1.2016

[25] H. tan, M. ren, B. liu, and J. song, "Pcl system with illuminator of opportunity," in *2006 CIE International Conference on Radar*, Oct 2006, pp. 1–3.

[26] M. E. V. Valkenburg, *Reference Data For Engineers*. Newnes, Boston, 2001.

[27] H. Van Trees, *Detection, Estimation, and Modulation Theory*. Wiley, 2004, no. Part 1.

[28] L. Wasserman, *All of Statistics: A Concise Course in Statistical Inference*, ser. Springer Texts in Statistics. Springer, 2004.

[29] R. Ziemer and W. Tranter, *Principles of Communications, 7th Edition:*. John Wiley & Sons, 2014.

APPENDIX A. INFORMATION MATRIX ELEMENTS

$$
\begin{aligned}
A_{11} =& a_{pq}\left(a_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau\tau}}\right)_{c,d}+e_{nm}\left(\boldsymbol{I}_{\boldsymbol{f\tau}}\right)_{c,d}+\nu_{im}\left(\boldsymbol{I}_{\boldsymbol{d_i\tau}}\right)_{m,d}/N+\eta_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_r\tau}}\right)_{n,d}/M\right)+\\
& e_{pq}\left(a_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau f}}\right)_{c,d}+e_{nm}\left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{cd}+\nu_{im}\left(\boldsymbol{I}_{\boldsymbol{d_if}}\right)_{m,d}/N+\eta_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rf}}\right)_{n,d}/M\right)+\\
& \nu_{iq}/N\left(a_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau d_i}}\right)_{c,q}+e_{nm}\left(\boldsymbol{I}_{\boldsymbol{fd_i}}\right)_{c,q}+\nu_{im}\left(\boldsymbol{I}_{\boldsymbol{d_id_i}}\right)_{m,q}/N+\eta_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rd_i}}\right)_{n,q}/M\right)+\\
& \eta_{rp}/M\left(a_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau d_r}}\right)_{c,p}+e_{nm}\left(\boldsymbol{I}_{\boldsymbol{fd_r}}\right)_{c,p}+\nu_{im}\left(\boldsymbol{I}_{\boldsymbol{d_id_r}}\right)_{m,p}/N+\eta_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rd_r}}\right)_{n,p}/M\right),
\end{aligned}
$$

$$
\begin{aligned}
A_{12} =& A_{21}=b_{pq}\left(a_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau\tau}}\right)_{c,d}+e_{nm}\left(\boldsymbol{I}_{\boldsymbol{f\tau}}\right)_{c,d}+\nu_{im}\left(\boldsymbol{I}_{\boldsymbol{d_i\tau}}\right)_{m,d}/N+\eta_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_r\tau}}\right)_{n,d}/M\right)+\\
& g_{pq}\left(a_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau f}}\right)_{c,d}+e_{nm}\left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{cd}+\nu_{im}\left(\boldsymbol{I}_{\boldsymbol{d_if}}\right)_{m,d}/N+\eta_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rf}}\right)_{n,d}/M\right)+\\
& \iota_{iq}/N\left(a_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau d_i}}\right)_{c,q}+e_{nm}\left(\boldsymbol{I}_{\boldsymbol{fd_i}}\right)_{c,q}+\nu_{im}\left(\boldsymbol{I}_{\boldsymbol{d_id_i}}\right)_{m,q}/N+\eta_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rd_i}}\right)_{n,q}/M\right)+\\
& \psi_{rp}/M\left(a_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau d_r}}\right)_{c,p}+e_{nm}\left(\boldsymbol{I}_{\boldsymbol{fd_r}}\right)_{c,p}+\nu_{im}\left(\boldsymbol{I}_{\boldsymbol{d_id_r}}\right)_{m,p}/N+\eta_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rd_r}}\right)_{n,p}/M\right),
\end{aligned}
$$

$$
\begin{aligned}
A_{13} =& A_{31}=c_{pq}\left(a_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau\tau}}\right)_{c,d}+e_{nm}\left(\boldsymbol{I}_{\boldsymbol{f\tau}}\right)_{c,d}+\nu_{im}\left(\boldsymbol{I}_{\boldsymbol{d_i\tau}}\right)_{m,d}/N+\eta_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_r\tau}}\right)_{n,d}/M\right)+\\
& h_{pq}\left(a_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau f}}\right)_{c,d}+e_{nm}\left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{cd}+\nu_{im}\left(\boldsymbol{I}_{\boldsymbol{d_if}}\right)_{m,d}/N+\eta_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rf}}\right)_{n,d}/M\right)+\\
& \rho_{iq}/N\left(a_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau d_i}}\right)_{c,q}+e_{nm}\left(\boldsymbol{I}_{\boldsymbol{fd_i}}\right)_{c,q}+\nu_{im}\left(\boldsymbol{I}_{\boldsymbol{d_id_i}}\right)_{m,q}/N+\eta_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rd_i}}\right)_{n,q}/M\right)+\\
& \omega_{rp}/M\left(a_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau d_r}}\right)_{c,p}+e_{nm}\left(\boldsymbol{I}_{\boldsymbol{fd_r}}\right)_{c,p}+\nu_{im}\left(\boldsymbol{I}_{\boldsymbol{d_id_r}}\right)_{m,p}/N+\eta_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rd_r}}\right)_{n,p}/M\right),
\end{aligned}
$$

$$
A_{14} = A_{41}=\beta_{pq}\left(a_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau f}}\right)_{c,d}+e_{nm}\left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{c,d}+\nu_{im}\left(\boldsymbol{I}_{\boldsymbol{d_if}}\right)_{m,d}/N+\eta_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rf}}\right)_{n,d}/M\right),
$$

$$
A_{15} = A_{51}=\kappa_{pq}\left(a_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau f}}\right)_{c,d}+e_{nm}\left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{c,d}+\nu_{im}\left(\boldsymbol{I}_{\boldsymbol{d_if}}\right)_{m,d}/N+\eta_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rf}}\right)_{n,d}/M\right),
$$

$$
A_{16} = A_{61}=\mu_{pq}\left(a_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau f}}\right)_{c,d}+e_{nm}\left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{c,d}+\nu_{im}\left(\boldsymbol{I}_{\boldsymbol{d_if}}\right)_{m,d}/N+\eta_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rf}}\right)_{n,d}/M\right),
$$

$$
\begin{aligned}
A_{22} =& b_{pq}\left(b_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau\tau}}\right)_{c,d}+g_{nm}\left(\boldsymbol{I}_{\boldsymbol{f\tau}}\right)_{c,d}+\iota_{im}\left(\boldsymbol{I}_{\boldsymbol{d_i\tau}}\right)_{m,d}/N+\psi_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_r\tau}}\right)_{n,d}/M\right)+\\
& g_{pq}\left(b_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau f}}\right)_{c,d}+g_{nm}\left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{cd}+\iota_{im}\left(\boldsymbol{I}_{\boldsymbol{d_if}}\right)_{m,d}/N+\psi_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rf}}\right)_{n,d}/M\right)+\\
& \iota_{iq}/N\left(b_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau d_i}}\right)_{c,q}+g_{nm}\left(\boldsymbol{I}_{\boldsymbol{fd_i}}\right)_{c,q}+\iota_{im}\left(\boldsymbol{I}_{\boldsymbol{d_id_i}}\right)_{m,q}/N+\psi_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rd_i}}\right)_{n,q}/M\right)+\\
& \psi_{rp}/M\left(b_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau d_r}}\right)_{c,p}+g_{nm}\left(\boldsymbol{I}_{\boldsymbol{fd_r}}\right)_{c,p}+\iota_{im}\left(\boldsymbol{I}_{\boldsymbol{d_id_r}}\right)_{m,p}/N+\psi_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rd_r}}\right)_{n,p}/M\right),
\end{aligned}
$$

$$
\begin{aligned}
A_{23} = A_{32} = {} & b_{pq}\left(c_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau\tau}}\right)_{c,d} + h_{nm}\left(\boldsymbol{I}_{\boldsymbol{f\tau}}\right)_{c,d} + \rho_{im}\left(\boldsymbol{I}_{\boldsymbol{d_i\tau}}\right)_{m,d}/N + \omega_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_r\tau}}\right)_{n,d}/M\right) + \\
& g_{pq}\left(c_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau f}}\right)_{c,d} + h_{nm}\left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{cd} + \rho_{im}\left(\boldsymbol{I}_{\boldsymbol{d_if}}\right)_{m,d}/N + \omega_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rf}}\right)_{n,d}/M\right) + \\
& \iota_{iq}/N\left(c_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau d_i}}\right)_{c,q} + h_{nm}\left(\boldsymbol{I}_{\boldsymbol{fd_i}}\right)_{c,q} + \rho_{im}\left(\boldsymbol{I}_{\boldsymbol{d_id_i}}\right)_{m,q}/N + \omega_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rd_i}}\right)_{n,q}/M\right) + \\
& \psi_{rp}/M\left(c_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau d_r}}\right)_{c,p} + h_{nm}\left(\boldsymbol{I}_{\boldsymbol{fd_r}}\right)_{c,p} + \rho_{im}\left(\boldsymbol{I}_{\boldsymbol{d_id_r}}\right)_{m,p}/N + \omega_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rd_r}}\right)_{n,p}/M\right),
\end{aligned}
$$

$$
A_{24} = A_{42} = \beta_{pq}\left(b_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau f}}\right)_{c,d} + g_{nm}\left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{c,d} + \iota_{im}\left(\boldsymbol{I}_{\boldsymbol{d_if}}\right)_{m,d}/N + \psi_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rf}}\right)_{n,d}/M\right),
$$

$$
A_{25} = A_{52} = \kappa_{pq}\left(b_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau f}}\right)_{c,d} + g_{nm}\left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{c,d} + \iota_{im}\left(\boldsymbol{I}_{\boldsymbol{d_if}}\right)_{m,d}/N + \psi_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rf}}\right)_{n,d}/M\right),
$$

$$
A_{26} = A_{62} = \mu_{pq}\left(b_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau f}}\right)_{c,d} + g_{nm}\left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{c,d} + \iota_{im}\left(\boldsymbol{I}_{\boldsymbol{d_if}}\right)_{m,d}/N + \psi_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rf}}\right)_{n,d}/M\right),
$$

$$
\begin{aligned}
A_{33} = {} & c_{pq}\left(c_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau\tau}}\right)_{c,d} + h_{nm}\left(\boldsymbol{I}_{\boldsymbol{f\tau}}\right)_{c,d} + \rho_{im}\left(\boldsymbol{I}_{\boldsymbol{d_i\tau}}\right)_{m,d}/N + \omega_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_r\tau}}\right)_{n,d}/M\right) + \\
& h_{pq}\left(c_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau f}}\right)_{c,d} + h_{nm}\left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{cd} + \rho_{im}\left(\boldsymbol{I}_{\boldsymbol{d_if}}\right)_{m,d}/N + \omega_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rf}}\right)_{n,d}/M\right) + \\
& \rho_{iq}/N\left(c_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau d_i}}\right)_{c,q} + h_{nm}\left(\boldsymbol{I}_{\boldsymbol{fd_i}}\right)_{c,q} + \rho_{im}\left(\boldsymbol{I}_{\boldsymbol{d_id_i}}\right)_{m,q}/N + \omega_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rd_i}}\right)_{n,q}/M\right) + \\
& \omega_{rp}/M\left(c_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau d_r}}\right)_{c,p} + h_{nm}\left(\boldsymbol{I}_{\boldsymbol{fd_r}}\right)_{c,p} + \rho_{im}\left(\boldsymbol{I}_{\boldsymbol{d_id_r}}\right)_{m,p}/N + \omega_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rd_r}}\right)_{n,p}/M\right),
\end{aligned}
$$

$$
A_{34} = A_{34} = \beta_{pq}\left(c_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau f}}\right)_{c,d} + h_{nm}\left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{c,d} + \rho_{im}\left(\boldsymbol{I}_{\boldsymbol{d_if}}\right)_{m,d}/N + \omega_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rf}}\right)_{n,d}/M\right),
$$

$$
A_{35} = A_{54} = \kappa_{pq}\left(c_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau f}}\right)_{c,d} + h_{nm}\left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{c,d} + \rho_{im}\left(\boldsymbol{I}_{\boldsymbol{d_if}}\right)_{m,d}/N + \omega_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rf}}\right)_{n,d}/M\right),
$$

$$
A_{36} = A_{64} = \mu_{pq}\left(c_{nm}\left(\boldsymbol{I}_{\boldsymbol{\tau f}}\right)_{c,d} + h_{nm}\left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{c,d} + \rho_{im}\left(\boldsymbol{I}_{\boldsymbol{d_if}}\right)_{m,d}/N + \omega_{rn}\left(\boldsymbol{I}_{\boldsymbol{d_rf}}\right)_{n,d}/M\right),
$$

$$
A_{44} = \beta_{pq}\beta_{nm}\left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{c,d},
$$

$$
A_{45} = A_{54} = \kappa_{pq}\beta_{nm}\left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{c,d},
$$

$$
A_{46} = A_{64} = \mu_{pq}\beta_{nm}\left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{c,d},
$$

$$
A_{55} = \kappa_{pq}\kappa_{nm}\left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{c,d},
$$

$$A_{56} = A_{65} = \mu_{pq}\kappa_{nm} \left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{c,d},$$

and

$$A_{66} = \mu_{pq}\mu_{nm} \left(\boldsymbol{I}_{\boldsymbol{ff}}\right)_{c,d},$$

where the multipliers $a_{nm}, b_{nm}, c_{nm}, e_{nm}, g_{nm}, h_{nm}, \beta_{nm}, \kappa_{nm}, \mu_{nm}, \nu_{im}, \iota_{im}, \rho_{im}, \eta_{rn}, \psi_{rn}$ and ω_{rn} are as denoted in Section 3.1. $\left(\boldsymbol{I}_{\boldsymbol{\vartheta_i\vartheta_j}}\right)_{a,b}$ is the (a, b)-th element of the respective block matrix of $\boldsymbol{I}\left(\boldsymbol{\vartheta}|\boldsymbol{\alpha}\right)$ and indices $c = m + M\left(n - 1\right)$ and $d = q + M\left(p - 1\right)$.

APPENDIX B. MATLAB SCRIPTS

```
function [ results ] = m_FIM(illuminators, receivers, targets)
        % These scripts are provided as is.
        % No significant effort has been made to verify the computational optimality.
        % Instead the focus was to make highly modular and easily modified structure
        % to calculate the joint Cramer-Rao lower bound of velocity and position
        % estimations.
        % Reference indices [n] in the code point to equations in the thesis.
        % Reference indices (n) in the code point to "Generalized Cramer-Rao Bound for
        % Joint Estimation of Target Position and Velocity for Active and Passive Radar Networks"
        % By Q. He, J. Hu, R. S. Blum and Y. Wu

        % Inputs
        %       illuminators: M x 1 Illuminator object array
        %       receivers: N x 1 Receiver object array
        %       targets: Target object column array of spesified size
        % Outputs
        %       results: l_t x 8 matrix, containing target x and y coordinates and
        %       parameterwise MSEs

    % INTIALIZATION
    [l_r,~] = size(receivers);
    [l_i,~] = size(illuminators);
        [l_t,~] = size(targets);
    coords_i = zeros(l_i,3);
    coords_r = zeros(l_r,3);
    for i = 1:l_i
        coords_i(i,:) = illuminators(i).Coordinates;
    end
    for i = 1:l_r
        coords_r(i,:) = receivers(i).Coordinates;
    end

    % MAIN LOOP
    % Calculate the FIM for all targets in given topology
    parfor index = 1:l_t

        target = targets(index);
        coord_t = target.Coordinates;

        N = length(receivers);
        M = length(illuminators);

        % INTERMEDIATE MATRICES
        % Calculate the matrices used to form the FIM of the intermediate
        % parameter vector \vartheta [22]
        % Matrices Sf, St, Stau and Sr are described in [88], [91], [79] and [99]
        % respectively

        R = m_R(coords_i, coords_r, coord_t);
        S = m_S(illuminators, receivers, target);
        Q = m_Q(receivers);

```

```
53          C = S*R*transpose(conj(S)) + Q;
54          Y = R*transpose(conj(S))/C;
55          Stau = m_Stau(illuminators, receivers, target);
56          Sf = m_Sf(illuminators, receivers, target);
57          St = m_St(illuminators, receivers, target);
58          Sr = m_Sr(illuminators, receivers, target);
59
60          % FINAL MATRICES
61          % Calculate the block matrices forming the FIM of the intermediate
62          % parameter vector [73]
63          J_tautau = m_J_tautau(R, S, C, Stau, Y);
64          J_dtdt = m_J_dtdt(R, S, C, N, M, St);
65          J_drdr = m_J_drdr(R, S, C, N, M, Sr);
66          J_ff = m_J_ff(R, S, C, Sf, Y);
67          J_tauf = m_J_tauf(R, S, C, Sf, Stau, Y);
68          J_ftau = transpose(conj(J_tauf));
69          J_dttau = m_J_dttau(R, S, C, N, M, St, Stau);
70          J_taudt = transpose(conj(J_dttau));
71          J_dtf = m_J_dttau(R, S, C, N, M, St, Sf);
72          J_fdt = transpose(conj(J_dtf));
73          J_drtau = m_J_drtau(R, S, C, N, M, Stau, Sr);
74          J_taudr = transpose(conj(J_drtau));
75          J_drf = m_J_drf(R, S, C, N, M, Sf, Sr);
76          J_fdr = transpose(conj(J_drf));
77          J_drdt = m_J_drdt(R, S, C, N, M, Sr, St);
78          J_dtdr = transpose(conj(J_drdt));
79
80          % MULTIPLIER MATRICES
81          % Calculate the illuminator and receiver specific derivatives a, b, c,
82          % e, g, h, \beta, \kappa, \mu, \nu, \iota, \rho, \eta, \psi and \omega
83          % of which the first two dimensions are discribed in [30-39]
84          [F, G, D_i, D_r, H] = m_delv(illuminators, receivers, target);
85
86          %% FISHER INFORMATION MATRIX
87          % Preallocate a zero matrix
88          I = zeros(6);
89          % Loop over four indices described in [44]
90          for p = 1:N
91              for q = 1:M
92                  for n = 1:N
93                      for m = 1:M
94                          c = M*(n-1)+m;
95                          d = M*(p-1)+q;
96
97                          % The indices of the derivative multipliers used in
98                          % final FIM formation
99                          %{
100                         \a_nm = F(1,(n-1)*M+m);
101                         \b_nm = F(2,(n-1)*M+m);
102                         \c_nm = F(3,(n-1)*M+m);
103                         \e_nm = G(1,(n-1)*M+m);
104                         \f_nm = G(2,(n-1)*M+m);
105                         \h_nm = G(3,(n-1)*M+m);
106                         \beta_nm = H(1,(n-1)*M+m);
107                         \kappa_nm = H(2,(n-1)*M+m);
108                         \mu_nm = H(3,(n-1)*M+m);
109                         \nu_m = D_i(1,m);
110                         \iota_m = D_i(2,m);
111                         \rho_m = D_i(3,m);
```

```
112                 \eta_n = D_r(1,n);
113                 \psi_n = D_r(2,n);
114                 \omega_n = D_r(3,n);
115                 %}
116
117                 % Formation of elements of FIM of \theta. Elementwise
118                 % description can be found at [45-54]
119                 I(1,1) = I(1,1) + ...
120                 F(1,(p-1)*M+q) * ( F(1,(n-1)*M+m)*J_tautau(c,d) + G(1,(n-1)
                        *M+m)*J_ftau(c,d) ...
121                                 + D_i(1,m)*J_dttau(m,d)/n + D_r(1,n)*
                                        J_drtau(n,d)/M ) ...
122                 + G(1,(p-1)*M+q) * ( F(1,(n-1)*M+m)*J_tauf(c,d) + G(1,(n-1)
                        *M+m)*J_ff(c,d) ...
123                                 + D_i(1,m)*J_dtf(m,d)/N + D_r(1,n)*J_drf(n,
                                        d)/M ) ...
124                 + D_i(1,q)/N * ( F(1,(n-1)*M+m)*J_taudt(c,q) + G(1,(n-1)*M+
                        m)*J_fdt(c,q) ...
125                                 + D_i(1,m)*J_dtdt(m,q)/N + D_r(1,n)*J_drdt(
                                        n,q)/M ) ...
126                 + D_r(1,p)/M * ( F(1,(n-1)*M+m)*J_taudr(c,p) + G(1,(n-1)*M+
                        m)*J_fdr(c,p) ...
127                                 + D_i(1,m)*J_dtdr(m,p)/N + D_r(1,n)*J_drdr(
                                        n,p)/M ) ;
128
129
130                 I(1,2) = I(1,2) + ...
131                 F(2,(p-1)*M+q) * ( F(1,(n-1)*M+m)*J_tautau(c,d) + G(1,(n-1)
                        *M+m)*J_ftau(c,d) ...
132                                 + D_i(1,m)*J_dttau(m,d)/n + D_r(1,n)*
                                        J_drtau(n,d)/M ) ...
133                 + G(2,(p-1)*M+q) * ( F(1,(n-1)*M+m)*J_tauf(c,d) + G(1,(n-1)
                        *M+m)*J_ff(c,d) ...
134                                 + D_i(1,m)*J_dtf(m,d)/N + D_r(1,n)*J_drf(n,
                                        d)/M ) ...
135                 + D_i(2,q)/N * ( F(1,(n-1)*M+m)*J_taudt(c,q) + G(1,(n-1)*M+
                        m)*J_fdt(c,q) ...
136                                 + D_i(1,m)*J_dtdt(m,q)/N + D_r(1,n)*J_drdt(
                                        n,q)/M ) ...
137                 + D_r(2,p)/M * ( F(1,(n-1)*M+m)*J_taudr(c,p) + G(1,(n-1)*M+
                        m)*J_fdr(c,p) ...
138                                 + D_i(1,m)*J_dtdr(m,p)/N + D_r(1,n)*J_drdr(
                                        n,p)/M ) ;
139                 I(2,1) = I(1,2);
140
141
142                 I(1,3) = I(1,3) + ...
143                 F(3,(p-1)*M+q) * ( F(1,(n-1)*M+m)*J_tautau(c,d) + G(1,(n-1)
                        *M+m)*J_ftau(c,d) ...
144                                 + D_i(1,m)*J_dttau(m,d)/n + D_r(1,n)*
                                        J_drtau(n,d)/M ) ...
145                 + G(3,(p-1)*M+q) * ( F(1,(n-1)*M+m)*J_tauf(c,d) + G(1,(n-1)
                        *M+m)*J_ff(c,d) ...
146                                 + D_i(1,m)*J_dtf(m,d)/N + D_r(1,n)*J_drf(n,
                                        d)/M ) ...
147                 + D_i(3,q)/N * ( F(1,(n-1)*M+m)*J_taudt(c,q) + G(1,(n-1)*M+
                        m)*J_fdt(c,q) ...
148                                 + D_i(1,m)*J_dtdt(m,q)/N + D_r(1,n)*J_drdt(
                                        n,q)/M ) ...
```

```
149                 + D_r(3,p)/M * ( F(1,(n-1)*M+m)*J_taudr(c,p) + G(1,(n-1)*M+
                        m)*J_fdr(c,p) ...
150                         + D_i(1,m)*J_dtdr(m,p)/N + D_r(1,n)*J_drdr(
                                n,p)/M ) ;
151                 I(3,1) = I(1,3);
152
153
154                 I(1,4) = I(1,4) + H(1,(p-1)*M+q) * ( F(1,(n-1)*M+m)*J_tauf(
                        c,d) ...
155                         + G(1,(n-1)*M+m)*J_ff(c,d) + D_i(1,m)*J_dtf
                                (m,d)/N ...
156                         + D_r(1,n)*J_drf(n,d)/M ) ;
157                 I(4,1) = I(1,4);
158
159
160                 I(1,5) = I(1,5) + H(2,(p-1)*M+q) * ( F(1,(n-1)*M+m)*J_tauf(
                        c,d) ...
161                         + G(1,(n-1)*M+m)*J_ff(c,d) + D_i(1,m)*J_dtf
                                (m,d)/N ...
162                         + D_r(1,n)*J_drf(n,d)/M ) ;
163                 I(5,1) = I(1,5);
164
165
166                 I(1,6) = I(1,6) + H(3,(p-1)*M+q) * ( F(1,(n-1)*M+m)*J_tauf(
                        c,d) ...
167                         + G(1,(n-1)*M+m)*J_ff(c,d) + D_i(1,m)*J_dtf
                                (m,d)/N ...
168                         + D_r(1,n)*J_drf(n,d)/M ) ;
169                 I(6,1) = I(1,6);
170
171
172                 I(2,2) = I(2,2) + ...
173                 F(2,(p-1)*M+q) * ( F(2,(n-1)*M+m)*J_tautau(c,d) + G(2,(n-1)
                        *M+m)*J_ftau(c,d) ...
174                         + D_i(2,m)*J_dttau(m,d)/n + D_r(2,n)*
                                J_drtau(n,d)/M ) ...
175                 + G(2,(p-1)*M+q) * ( F(2,(n-1)*M+m)*J_tauf(c,d) + G(2,(n-1)
                        *M+m)*J_ff(c,d) ...
176                         + D_i(2,m)*J_dtf(m,d)/N + D_r(2,n)*J_drf(n,
                                d)/M ) ...
177                 + D_i(2,q)/N * ( F(2,(n-1)*M+m)*J_taudt(c,q) + G(2,(n-1)*M+
                        m)*J_fdt(c,q) ...
178                         + D_i(2,m)*J_dtdt(m,q)/N + D_r(2,n)*J_drdt(
                                n,q)/M ) ...
179                 + D_r(2,p)/M * ( F(2,(n-1)*M+m)*J_taudr(c,p) + G(2,(n-1)*M+
                        m)*J_fdr(c,p) ...
180                         + D_i(2,m)*J_dtdr(m,p)/N + D_r(2,n)*J_drdr(
                                n,p)/M ) ;
181
182
183                 I(2,3) = I(2,3) + ...
184                 F(3,(p-1)*M+q) * ( F(2,(n-1)*M+m)*J_tautau(c,d) + G(2,(n-1)
                        *M+m)*J_ftau(c,d) ...
185                         + D_i(2,m)*J_dttau(m,d)/n + D_r(2,n)*
                                J_drtau(n,d)/M ) ...
186                 + G(3,(p-1)*M+q) * ( F(2,(n-1)*M+m)*J_tauf(c,d) + G(2,(n-1)
                        *M+m)*J_ff(c,d) ...
187                         + D_i(2,m)*J_dtf(m,d)/N + D_r(2,n)*J_drf(n,
                                d)/M ) ...
```

```
188                 + D_i(3,q)/N * ( F(2,(n-1)*M+m)*J_taudt(c,q) + G(2,(n-1)*M+
                        m)*J_fdt(c,q) ...
189                     + D_i(2,m)*J_dtdt(m,q)/N + D_r(2,n)*J_drdt(
                            n,q)/M ) ...
190                 + D_r(3,p)/M * ( F(2,(n-1)*M+m)*J_taudr(c,p) + G(2,(n-1)*M+
                        m)*J_fdr(c,p) ...
191                     + D_i(2,m)*J_dtdr(m,p)/N + D_r(2,n)*J_drdr(
                            n,p)/M ) ;
192                 I(3,2) = I(2,3);
193
194
195                 I(2,4) = I(2,4) + H(1,(p-1)*M+q) * ( F(2,(n-1)*M+m)*J_tauf(
                        c,d) ...
196                     + G(2,(n-1)*M+m)*J_ff(c,d) + D_i(2,m)*J_dtf
                            (m,d)/N ...
197                     + D_r(2,n)*J_drf(n,d)/M ) ;
198                 I(4,2) = I(2,4);
199
200
201                 I(2,5) = I(2,5) + H(2,(p-1)*M+q) * ( F(2,(n-1)*M+m)*J_tauf(
                        c,d) ...
202                     + G(2,(n-1)*M+m)*J_ff(c,d) + D_i(2,m)*J_dtf
                            (m,d)/N ...
203                     + D_r(2,n)*J_drf(n,d)/M ) ;
204                 I(5,2) = I(2,5);
205
206
207                 I(2,6) = I(2,6) + H(3,(p-1)*M+q) * ( F(2,(n-1)*M+m)*J_tauf(
                        c,d) ...
208                     + G(2,(n-1)*M+m)*J_ff(c,d) + D_i(2,m)*J_dtf
                            (m,d)/N ...
209                     + D_r(2,n)*J_drf(n,d)/M ) ;
210                 I(6,2) = I(2,6);
211
212
213                 I(3,3) = I(3,3) + ...
214                 F(3,(p-1)*M+q) * ( F(3,(n-1)*M+m)*J_tautau(c,d) + G(3,(n-1)
                        *M+m)*J_ftau(c,d) ...
215                     + D_i(3,m)*J_dttau(m,d)/n + D_r(3,n)*
                            J_drtau(n,d)/M ) ...
216                 + G(3,(p-1)*M+q) * ( F(3,(n-1)*M+m)*J_tauf(c,d) + G(3,(n-1)
                        *M+m)*J_ff(c,d) ...
217                     + D_i(3,m)*J_dtf(m,d)/N + D_r(3,n)*J_drf(n,
                            d)/M ) ...
218                 + D_i(3,q)/N * ( F(3,(n-1)*M+m)*J_taudt(c,q) + G(3,(n-1)*M+
                        m)*J_fdt(c,q) ...
219                     + D_i(3,m)*J_dtdt(m,q)/N + D_r(3,n)*J_drdt(
                            n,q)/M ) ...
220                 + D_r(3,p)/M * ( F(3,(n-1)*M+m)*J_taudr(c,p) + G(3,(n-1)*M+
                        m)*J_fdr(c,p) ...
221                     + D_i(3,m)*J_dtdr(m,p)/N + D_r(3,n)*J_drdr(
                            n,p)/M ) ;
222
223
224                 I(3,4) = I(3,4) + H(1,(p-1)*M+q) * ( F(3,(n-1)*M+m)*J_tauf(
                        c,d) ...
225                     + G(3,(n-1)*M+m)*J_ff(c,d) + D_i(3,m)*J_dtf
                            (m,d)/N ...
226                     + D_r(3,n)*J_drf(n,d)/M ) ;
```

```
227                         I(4,3) = I(3,4);
228
229
230                         I(3,5) = I(3,5) + H(2,(p-1)*M+q) * ( F(3,(n-1)*M+m)*J_tauf(
                                c,d) ...
231                                     + G(3,(n-1)*M+m)*J_ff(c,d) + D_i(3,m)*J_dtf
                                        (m,d)/N ...
232                                     + D_r(3,n)*J_drf(n,d)/M ) ;
233                         I(5,3) = I(3,5);
234
235
236                         I(3,6) = I(3,6) + H(3,(p-1)*M+q) * ( F(3,(n-1)*M+m)*J_tauf(
                                c,d) ...
237                                     + G(3,(n-1)*M+m)*J_ff(c,d) + D_i(3,m)*J_dtf
                                        (m,d)/N ...
238                                     + D_r(3,n)*J_drf(n,d)/M ) ;
239                         I(6,3) = I(3,6);
240
241
242                         I(4,4) = I(4,4) + H(1,(p-1)*M+q)*H(1,(n-1)*M+m)*J_ff(c,d);
243
244
245                         I(4,5) = I(4,5) + H(2,(p-1)*M+q)*H(1,(n-1)*M+m)*J_ff(c,d);
246                         I(5,4) = I(4,5);
247
248
249                         I(4,6) = I(4,6) + H(3,(p-1)*M+q)*H(1,(n-1)*M+m)*J_ff(c,d);
250                         I(6,4) = I(4,6);
251
252
253                         I(5,5) = I(5,5) + H(2,(p-1)*M+q)*H(2,(n-1)*M+m)*J_ff(c,d);
254
255
256                         I(5,6) = I(5,6) + H(3,(p-1)*M+q)*H(2,(n-1)*M+m)*J_ff(c,d);
257                         I(6,5) = I(5,6);
258
259
260                         I(6,6) = I(6,6) + H(3,(p-1)*M+q)*H(3,(n-1)*M+m)*J_ff(c,d);
261                     end
262                 end
263             end
264         end
265
266                 % Some values may be rather close to singularity. Much of these are
                        with one of few
267                 % illuminators, and really small variances for velocity components.
268                 % Some near-singular values can be removed, if one is only
                        interested in position estimation.
269         results(index,:) = [coord_t(1),coord_t(2),diag(pinv(I)).'];
270     end
271 end
```

```
1  function [ S ] = m_S(illuminators, receivers, target)
2  % Inputs
3  %   illuminators: M x 1 Illuminator object array
4  %   receivers: N x 1 Receiver object array
5  %   targets: ~ x 1 Target object array of spesified size
6  % Outputs
7  %   S: The matrix containig all time delayed and doppler shifted signals from all
       illuminator-receiver pairs over all time samples. (13)
```

```
8
9
10 % Initiate necessary indices, constants and coordinate vector.
11 M = length(illuminators);
12 N = length(receivers);
13 K = Constants.K;
14 T_s = Constants.T_s;
15 P_0 = Constants.P_0;
16 c_t = target.Coordinates;
17 S = [];
18
19 % Loop over all illuminator-receiver pairs and all time samples.
20 for n = 1:N
21     % Initiate receiver specific values.
22     c_r = receivers(n).Coordinates;
23     % Preallocate a zero matrix of appropriate size.
24     U_n = zeros(K,M);
25     for k = 1:K
26         % Preallocate a zero column vector of appropriate size.
27         u_nk = zeros(M,1);
28         for m = 1:M
29             % Initiate illuminator specific values.
30             c_i = illuminators(m).Coordinates;
31             E_m = illuminators(m).Transmit_energy;
32             var = k*T_s-par_tau(c_t,c_i,c_r);
33             u_nmk = sqrt((E_m*P_0)/(norm(c_r-c_t)^2*norm(c_i-c_t)^2)) ...
34                     * exp(1i*2*pi*par_f(target,illuminators(m),receivers(n))*k*T_s)
                           ...
35                     * illuminators(m).Normalization_factor * d_non(var, Wave(
                           illuminators(m)));
36             u_nk(m) = u_nmk;
37         end
38         U_n(k,:) = (u_nk);
39     end
40     S = blkdiag(S,U_n);
41 end
```

```
1  function [ Q ] = m_Q(receivers)
2  % Inputs
3  %   receivers: N x 1 Receiver object array
4  % Outputs
5  %   Q: The matrix containig complex Gaussian clutter plus noise of all illuminator-
       receiver pairs. (70)
6
7  %% PARAMETERS:
8  % Exponential decay in corelation with distance
9  gamma = 0.000005;
10 % Variance of complex gaussian distributed clutter-plus-noise
11 variance = 0.9;
12
13 % NOTE: Current implementation yields spatially dependent clutter plus
14 % noise.
15
16 % Initialize necessary indices and constants
17 N = length(receivers);
18 K = Constants.K;
19
20 % Preallocate a zero matrix of appropriate size.
21 Q_tilde = zeros(N,N);
22
```

```
% To my knowledge, this could be done without loop
for n1 = 1:N
    c_r1 = receivers(n1).Coordinates;
    for n2 = 1:N
        c_r2 = receivers(n2).Coordinates;
        Q_tilde(n1,n2) = exp(-norm(c_r1-c_r2)*gamma);
    end
end

Q = kron(variance*Q_tilde,eye(K));
```

```
function [ R ] = m_R(c_illuminators, c_receivers, c_target)
% Inputs
%   c_illuminators: M x 3 matrix of illuminator coordinates
%   c_receivers: N x 3 matrix of receiver coordinates
%   c_targets: ~ x 3 spesified size matrix of target coordinates
% Outputs
%   R: The matrix containig complex Gaussian reflection coefficients of all
    illuminator-receiver pairs. (65)

%% PARAMETERS:
% Exponential decay in correlation with angle
omega = 0.01;

% NOTE: Current implementation yields spatially dependent coefficients.

% Initialize the number of receivers and illuminators.
[M,~] = size(c_illuminators);
[N,~] = size(c_receivers);

% Calculate the angle matrix between all receiver-target paths
v = c_receivers'-c_target'*ones(1,N);
angles_nn = acos((v'*v)./(sqrt(diag(v'*v)*(diag(v'*v))')));
angles_r = exp(-omega*angles_nn);

% Calculate the angle matrix between all illuminator-target paths
v = c_illuminators'-c_target'*ones(1,M);
angles_mm = acos((v'*v)./(sqrt(diag(v'*v)*(diag(v'*v))')));
angles_i = exp(-omega*angles_mm);

R = kron(angles_r, angles_i);
```

```
function [ F, G, D_i, D_r, H ] = m_delv(illuminators, receivers, target)
% Calculates the block matrices of Del_\theta\vartheta [24] containing all
% interesting derivatives of intermediate parameters by target variables.
% Inputs
%   illuminators: M x 1 Illuminator object array
%   receivers: N x 1 Receiver object array
%   targets: ~ x 1 Target object array of spesified size
% Outputs
%   F: 3 x N*M derivative matrix in equation [3.23]
%   G: 3 x N*M derivative matrix in equation [3.24]
%   D_i: 3 x M derivative matrix in equation [3.26]
%   D_r: 3 x N derivative matrix in equation [3.27]
%   H: 3 x N*M derivative matrix in equation [3.25]

% Initialize static variables.
M = length(illuminators);
N = length(receivers);
c_t = target.Coordinates;
```

```
19 v_t = target.Velocities;
20
21 % Preallocate zero matrices of appropriate size.
22 F = zeros(3,N*M);
23 G = zeros(3,N*M);
24 H = zeros(3,N*M);
25 D_i = zeros(3,M);
26 D_r = zeros(3,N);
27
28 for n = 1:N
29     % Initialize receiver specific variables
30     c_r = receivers(n).Coordinates;
31     d_r = norm(c_t-c_r);
32     for m = 1:M
33         % Initialize illuminator specific variables
34         c_i = illuminators(m).Coordinates;
35         d_i = norm(c_t-c_i);
36
37         % Matrix F, \tau derivated by spatial variables
38         F(1,(n-1)*M+m) = 1/Constants.c*((c_t(1)-c_i(1)/d_i) + (c_t(1)-c_r(1)/d_r));
39         F(2,(n-1)*M+m) = 1/Constants.c*((c_t(2)-c_i(2)/d_i) + (c_t(2)-c_r(2)/d_r));
40         F(3,(n-1)*M+m) = 1/Constants.c*((c_t(3)-c_i(3)/d_i) + (c_t(3)-c_r(3)/d_r));
41
42         % Matrix G, f derivated by spatial variables
43         G(1,(n-1)*M+m) = -v_t(1)/Constants.lambda * (1/d_i + 1/d_r) ...
44             + ((c_i(1)-c_t(1))/(Constants.lambda*d_i^3)) * (v_t(1)*(c_i(1)-c_t(1))
                    + v_t(2)*(c_i(2)-c_t(2)) + v_t(3)*(c_i(3)-c_t(3))) ...
45             + ((c_r(1)-c_t(1))/(Constants.lambda*d_r^3)) * (v_t(1)*(c_r(1)-c_t(1))
                    + v_t(2)*(c_r(2)-c_t(2)) + v_t(3)*(c_i(3)-c_t(3)));
46         G(2,(n-1)*M+m) = -v_t(2)/Constants.lambda * (1/d_i + 1/d_r) ...
47             + ((c_i(2)-c_t(2))/(Constants.lambda*d_i^3)) * (v_t(1)*(c_i(1)-c_t(1))
                    + v_t(2)*(c_i(2)-c_t(2)) + v_t(3)*(c_i(3)-c_t(3))) ...
48             + ((c_r(2)-c_t(2))/(Constants.lambda*d_r^3)) * (v_t(1)*(c_r(1)-c_t(1))
                    + v_t(2)*(c_r(2)-c_t(2)) + v_t(3)*(c_r(3)-c_t(3)));
49         G(3,(n-1)*M+m) = -v_t(3)/Constants.lambda * (1/d_i + 1/d_r) ...
50             + ((c_i(3)-c_t(3))/(Constants.lambda*d_i^3)) * (v_t(1)*(c_i(1)-c_t(1))
                    + v_t(2)*(c_i(2)-c_t(2)) + v_t(3)*(c_i(3)-c_t(3))) ...
51             + ((c_r(3)-c_t(3))/(Constants.lambda*d_r^3)) * (v_t(1)*(c_r(1)-c_t(1))
                    + v_t(2)*(c_r(2)-c_t(2)) + v_t(3)*(c_r(3)-c_t(3)));
52
53         % Matrix H, f derivated by velocity variables
54         H(1,(n-1)*M+m) = (c_i(1)-c_t(1))/(Constants.lambda*d_i) + (c_r(1)-c_t(1))/(
               Constants.lambda*d_r);
55         H(2,(n-1)*M+m) = (c_i(2)-c_t(2))/(Constants.lambda*d_i) + (c_r(2)-c_t(2))/(
               Constants.lambda*d_r);
56         H(3,(n-1)*M+m) = (c_i(3)-c_t(3))/(Constants.lambda*d_i) + (c_r(3)-c_t(3))/(
               Constants.lambda*d_r);
57     end
58
59     % Matrix D_t, d_t derivated by spatial variables
60     D_r(1,n) = (c_t(1)-c_r(1))/d_r;
61     D_r(2,n) = (c_t(2)-c_r(2))/d_r;
62     D_r(3,n) = (c_t(3)-c_r(3))/d_r;
63 end
64
65 % This has a separate loop structure for enchanced readability without
66 % multiple and unnecessary computing.
67 for m = 1:M
68     % Initialize illuminator specific variables
```

```
69      c_i = illuminators(m).Coordinates;
70      d_i = norm(c_t-c_i);
71
72      % Matrix D_i, d_i derivated by velocity variables
73      D_i(1,m) = (c_t(1)-c_i(1))/d_i;
74      D_i(2,m) = (c_t(2)-c_i(2))/d_i;
75      D_i(3,m) = (c_t(3)-c_i(3))/d_i;
76  end
```

```
1   function [ J_drdr ] = m_J_drdr(R, S, C, N, M, Sr)
2   % Inputs
3   %   R: The matrix containig complex Gaussian reflection coefficients of illuminator
        -receiver paths
4   %   S: The matrix containig all time delayed and doppler shifted signal samples
        from all illuminator-receiver paths. (13)
5   %   C: Covariance matrix. (18)
6   %   N: Number of receivers in the system.
7   %   M: Number of illuminators in the system.
8   %   Sr: The intermediate matrix of the waveform derivates over receiver distances.
        (99)
9   % Outputs
10  %   J_drdr: A block matrix of FIM of \vartheta. (103)
11
12
13  S_1 = 0;
14  for m1 = 1:M
15      for m2 = 1:M
16          lambda_m1 = (transpose(conj(R(:,m1:m1+(N-1)*M))));
17          lambda_m2 = (transpose(conj(R(:,m2:m2+(N-1)*M))));
18          lambda_m1h = transpose(conj(lambda_m1*M));
19          p_m1 = Sr(:,m1:m1+(N-1)*M);
20          p_m2 = Sr(:,m2:m2+(N-1)*M);
21          p_m1h = transpose(conj(p_m1));
22          Sh = transpose(conj(S));
23          p1 = (lambda_m1*Sh/C*p_m2).*transpose(lambda_m2*Sh/C*p_m1);
24          p2 = (p_m1h/C*p_m2).*transpose(lambda_m2*Sh/C*S*lambda_m1h);
25
26          % Increment intermediate matrix.
27          S_1 = S_1 + p1 + p2;
28      end
29  end
30
31  J_drdr = 2*real(S_1);
```

```
1   function [ J_drdt ] = m_J_drdt(R, S, C, N, M, Sr, St)
2   % Inputs
3   %   R: The matrix containig complex Gaussian reflection coefficients of illuminator
        -receiver paths.
4   %   S: The matrix containig all time delayed and doppler shifted signal samples
        from all illuminator-receiver paths. (13)
5   %   C: Covariance matrix. (18)
6   %   N: Number of receivers in the system.
7   %   M: Number of illuminators in the system.
8   %   Sr: The intermediate matrix of the waveform derivates over receiver distances.
        (99)
9   %   St: The intermediate matrix of the waveform derivates over illuminator
        distances. (91)
10  % Outputs
11  %   J_drdt: A block matrix of FIM of \vartheta. (106)
12
```

```
13
14 S_1 = 0;
15 for m = 1:M
16     for n = 1:N
17         aleph_n = (transpose(conj(R(:,(1+(n-1)*M):M+(n-1)*M))));
18         I_n = St(:,(1+(n-1)*M):M+(n-1)*M);
19         lambda_m = (transpose(conj(R(:,m:m+(N-1)*M))));
20         lambda_mh = transpose(conj(lambda_m*M));
21         p_m = Sr(:,m:m+(N-1)*M);
22         p_mh = transpose(conj(p_m));
23         Sh = transpose(conj(S));
24         p1 = (lambda_m*Sh/C*I_n).*transpose(aleph_n*Sh/C*p_m);
25         p2 = (p_mh/C*I_n).*transpose(aleph_n*Sh/C*S*lambda_mh);
26
27         % Increment intermediate matrix
28         S_1 = S_1 + p1 + p2;
29     end
30 end
31
32 J_drdt = 2*real(S_1);
```

```
1  function [ J_drf ] = m_J_drf(R, S, C, N, M, Sf, Sr)
2  % Inputs
3  %   R: The matrix containig complex Gaussian reflection coefficients of illuminator
       -receiver paths.
4  %   S: The matrix containig all time delayed and doppler shifted signal samples
       from all illuminator-receiver paths. (13)
5  %   C: Covariance matrix. (18)
6  %   N: Number of receivers in the system.
7  %   M: Number of illuminators in the system.
8  %   Sf: The intermediate matrix of the waveform derivates over doppler shifts. (88)
9  %   Sr: The intermediate matrix of the waveform derivates over receiver distances.
       (99)
10 % Outputs
11 %   J_drf: A block matrix of FIM of \vartheta. (105)
12
13
14 % Notation and indexing of input matrices:
15 %s(i) = S(:,i)
16 %z(i) = R(:,i)
17
18 S_1 = 0;
19 for m = 1:M
20     lambda_m = transpose(conj(R(:,m:m+(N-1)*M)));
21     lambda_mh = transpose(conj(lambda_m));
22     p_m = Sr(:,m:m+(N-1)*M);
23     p_mh = transpose(conj(p_m));
24     Sh = transpose(conj(S));
25     p1 = (lambda_m*Sh/C*Sf).*transpose(R*Sh/C*p_m);
26     p2 = (p_mh/C*Sf).*transpose(R*Sh/C*S*lambda_mh);
27
28     % Increment intermediate matrix.
29     S_1 = S_1 + p1 + p2;
30 end
31
32 J_drf = 2*real(S_1);
```

```
1  function [ J_drtau ] = m_J_drtau(R, S, C, N, M, Stau, Sr)
2  % Inputs
```

```
3  %   R: The matrix containig complex Gaussian reflection coefficients of illuminator
       -receiver paths.
4  %   S: The matrix containig all time delayed and doppler shifted signal samples
       from all illuminator-receiver paths. (13)
5  %   C: Covariance matrix. (18)
6  %   N: Number of receivers in the system.
7  %   M: Number of illuminators in the system.
8  %   Stau: The intermediate matrix of the waveform derivates over time delays.
9  %   Sr: The intermediate matrix of the waveform derivates over receiver distances.
       (99)
10 % Outputs
11 %   J_drtau: A block matrix of FIM of \vartheta. (104)
12
13
14 % Notation and indexing of input matrices:
15 %s(i) = S(:,i)
16 %z(i) = R(:,i)
17
18 S_1 = 0;
19 for m = 1:M
20     lambda_m = transpose(conj(R(:,m:m+(N-1)*M)));
21     lambda_mh = transpose(conj(lambda_m));
22     p_m = Sr(:,m:m+(N-1)*M);
23     p_mh = transpose(conj(p_m));
24     Sh = transpose(conj(S));
25     p1 = (lambda_m*Sh/C*Stau).*transpose(R*Sh/C*p_m);
26     p2 = (p_mh/C*Stau).*transpose(R*Sh/C*S*lambda_mh);
27
28     % Increment intermediate matrix.
29     S_1 = S_1 + p1 + p2;
30 end
31
32 J_drtau = 2*real(S_1);
```

```
1  function [ J_dtdt ] = m_J_dtdt(R, S, C, N, M, St)
2  % Inputs
3  %   R: The matrix containig complex Gaussian reflection coefficients of illuminator
       -receiver paths.
4  %   S: The matrix containig all time delayed and doppler shifted signal samples
       from all illuminator-receiver paths. (13)
5  %   C: Covariance matrix. (18)
6  %   N: Number of receivers in the system.
7  %   M: Number of illuminators in the system.
8  %   St: The intermediate matrix of the waveform derivates over illuminator
       distances. (91)
9  % Outputs
10 %   J_dtdt: A block matrix of FIM of \vartheta. (95)
11
12
13 % Notation and indexing of input matrices:
14 % s(i) = S(:,i)
15 % z(i) = R(:,i)
16 % aleph(i) = (transpose(conj(R(:,(1+(i-1)*M):M+(i-1)*M))))
17 % I(i) = St(:,(1+(i-1)*M):M+(i-1)*M)
18
19 S_1 = 0;
20 for n1 = 1:N
21     for n2 = 1:N
22         aleph_n1 = (transpose(conj(R(:,(1+(n1-1)*M):M+(n1-1)*M))));
23         aleph_n2 = (transpose(conj(R(:,(1+(n2-1)*M):M+(n2-1)*M))));
```

```
        aleph_n1h = transpose(conj(aleph_n1));
        I_n1 = St(:,(1+(n1-1)*M):M+(n1-1)*M);
        I_n2 = St(:,(1+(n2-1)*M):M+(n2-1)*M);
        I_n1h = transpose(conj(I_n1));
        Sh = transpose(conj(S));
        p1 = (aleph_n1*Sh/C*I_n2).*transpose(aleph_n2*Sh/C*I_n1);
        p2 = (I_n1h/C*I_n2).*transpose(aleph_n2*Sh/C*S*aleph_n1h);

        % Increment intermediate matrix.
        S_1 = S_1 + p1 + p2;
    end
end

J_dtdt = 2*real(S_1);
```

```
function [ J_dtf ] = m_J_dtf(R, S, C, N, M, St, Sf)
% Inputs
%   R: The matrix containig complex Gaussian reflection coefficients of illuminator-receiver paths.
%   S: The matrix containig all time delayed and doppler shifted signal samples from all illuminator-receiver paths. (13)
%   C: Covariance matrix. (18)
%   N: Number of receivers in the system.
%   M: Number of illuminators in the system.
%   St: The intermediate matrix of the waveform derivates over illuminator distances. (91)
%   Sf: The intermediate matrix of the waveform derivates over doppler shifts. (88)
% Outputs
%   J_dtf: A block matrix of FIM of \vartheta. (97)

% Notation and indexing of input matrices:
% s(i) = S(:,i)
% z(i) = R(:,i)
% aleph(i) = (transpose(conj(R(:,(1+(i-1)*M):M+(i-1)*M))))
% I(i) = St(:,(1+(i-1)*M):M+(i-1)*M)

S_1 = 0;
for n = 1:N
    aleph_n = (transpose(conj(R(:,(1+(n-1)*M):M+(n-1)*M))));
    aleph_nh = transpose(conj(aleph_n));
    I_n = St(:,(1+(n-1)*M):M+(n-1)*M);
    I_nh = transpose(conj(I_n));
    Sh = transpose(conj(S));
    p1 = (aleph_n*Sh/C*Sf).*tranpose(R*Sh/C*I_n);
    p2 = (I_nh/C*Sf).*transpose(R*Sh/C*S*aleph_nh);

    % Increment intermediate matrix.
    S_1 = S_1 + p1 + p2;
end

J_dtf = 2*real(S_1);
```

```
function [ J_dttau ] = m_J_dttau(R, S, C, N, M, St, Stau)
% Inputs
%   R: The matrix containig complex Gaussian reflection coefficients of illuminator-receiver paths.
%   S: The matrix containig all time delayed and doppler shifted signal samples from all illuminator-receiver paths. (13)
%   C: Covariance matrix. (18)
%   N: Number of receivers in the system.
```

```
7  %   M: Number of illuminators in the system.
8  %   St: The intermediate matrix of the waveform derivates over illuminator
       distances. (91)
9  %   Stau: The intermediate matrix of the waveform derivates over time delays.
10 % Outputs
11 %   J_dttau: A block matrix of FIM of \vartheta. (96)
12
13
14 % Notation and indexing of input matrices:
15 % s(i) = S(:,i)
16 % z(i) = R(:,i)
17 % aleph(i) = (transpose(conj(R(:,(1+(i-1)*M):M+(i-1)*M))))
18 % I(i) = St(:,(1+(i-1)*M):M+(i-1)*M)
19
20 S_1 = 0;
21 for n = 1:N
22     aleph_n = (transpose(conj(R(:,(1+(n-1)*M):M+(n-1)*M))));
23     aleph_nh = transpose(conj(aleph_n));
24     I_n = St(:,(1+(n-1)*M):M+(n-1)*M);
25     I_nh = transpose(conj(I_n));
26     Sh = transpose(conj(S));
27     p1 = (aleph_n*Sh/C*Stau).*transpose(R*Sh/C*I_n);
28     p2 = (I_nh/C*Stau).*transpose(R*Sh/C*S*aleph_nh);
29
30     % Increment intermediate matrix.
31     S_1 = S_1 + p1 + p2;
32 end
33
34 J_dttau = 2*real(S_1);
```

```
1  function [ J_ff ] = m_J_ff(R, S, C, Sf, Y)
2  % Inputs
3  %   R: The matrix containig complex Gaussian reflection coefficients of illuminator
       -receiver paths.
4  %   S: The matrix containig all time delayed and doppler shifted signal samples
       from all illuminator-receiver paths. (13)
5  %   C: Covariance matrix. (18)
6  %   Sf: The intermediate matrix of the waveform derivates over doppler shifts. (88)
7  %   Y: An intermediate matrix combining R, S and C. (85-86)
8  % Outputs
9  %   J_ff: A block matrix of FIM of \vartheta. (87)
10
11
12 M_1 = (Y*Sf).*transpose(Y*Sf);
13 M_2 = (transpose(conj(Sf))/C*Sf).*(transpose(Y*S*R));
14 J_ff = 2*real(M_1 + M_2);
```

```
1  function [ J_tauf ] = m_J_tauf(R, S, C, Sf, Stau, Y)
2  % Inputs
3  %   R: The matrix containig complex Gaussian reflection coefficients of illuminator
       -receiver paths.
4  %   S: The matrix containig all time delayed and doppler shifted signal samples
       from all illuminator-receiver paths. (13)
5  %   C: Covariance matrix. (18)
6  %   Sf: The intermediate matrix of the waveform derivates over doppler shifts. (88)
7  %   Stau: The intermediate matrix of the waveform derivates over time delays.
8  %   Y: An intermediate matrix combining R, S and C. (85-86)
9  % Outputs
10 %   J_tauf: A block matrix of FIM of \vartheta. (86)
11
```

```
12
13 M_1 = (Y*Sf).*transpose(Y*Stau);
14 M_2 = (transpose(conj(Stau))/C*Sf).*(transpose(Y*S*R));
15 J_tauf = 2*real(M_1 + M_2);
```

```
1  function [ J_tautau ] = m_J_tautau(R, S, C, Stau, Y)
2  % Inputs
3  %   R: The matrix containig complex Gaussian reflection coefficients of illuminator
       -receiver paths.
4  %   S: The matrix containig all time delayed and doppler shifted signal samples
       from all illuminator-receiver paths. (13)
5  %   C: Covariance matrix. (18)
6  %   Stau: The intermediate matrix of the waveform derivates over time delays.
7  %   Y: An intermediate matrix combining R, S and C. (85-86)
8  % Outputs
9  %   J_tautau: A block matrix of FIM of \vartheta. (85)
10
11
12 M_1 = (Y*Stau).*transpose(Y*Stau);
13 M_2 = (transpose(conj(Stau))*(C\Stau)).*(transpose(Y*S*R));
14 J_tautau = 2*real(M_1 + M_2);
```

```
1  function [ Sf ] = m_Sf(illuminators, receivers, target)
2  % Inputs
3  %   illuminators: M x 1 Illuminator object array
4  %   receivers: N x 1 Receiver object array
5  %   target: Target object
6  % Outputs
7  %   Sf: The intermediate matrix of the waveform derivates over doppler shifts. (88)
8
9  % Initiate necessary indices
10 M = length(illuminators);
11 N = length(receivers);
12
13 % Preallocate a zero matrix of appropriate size
14 Sf = zeros(Constants.K*N,M*N);
15 % Loop over all illuminator-receiver pairs to generate elements
16 for n = 1:N
17     for m = 1:M
18         Sf(:,M*(n-1)+m) = m_Sfi(illuminators(m),receivers(n),target,n,N);
19     end
20 end
```

```
1  function [ Sfi ] = m_Sfi(illuminator, receiver, target, n, N)
2  % Inputs
3  %   illuminator: Illuminator object
4  %   receiver: Receiver object
5  %   target: Target object
6  %   n: The Index of receiver
7  %   N: Number of receivers in the system
8  % Outputs
9  %   Sfi: ith Column vector of Sf. (88)
10
11 % Initiate the parameters from object and known constants.
12 c_r = receiver.Coordinates;
13 c_i = illuminator.Coordinates;
14 c_t = target.Coordinates;
15 E_m = illuminator.Transmit_energy;
16 T_s = Constants.T_s;
17 K = Constants.K;
```

```
18 P_0 = Constants.P_0;
19
20 % Preallocate a zero column vector of appropriate size.
21 p = zeros(K,1);
22 % Loop over time samples .
23 for k = 1:K
24     p1 = sqrt((E_m*P_0)/(norm(c_r-c_t)^2*norm(c_i-c_t)^2)) * exp(1i*2*pi*par_f(
       target,illuminator,receiver)*k*T_s);
25     p2 = 1i*2*pi*k*T_s * illuminator.Normalization_factor * d_non(k*T_s,Wave(
       illuminator));
26     p(k) = p1 * p2;
27 end
28
29 % Reshape the product to the final column matrix.
30 e_n = full(sparse(n,1,1,N,1));
31 Sfi = kron(e_n,p);
```

```
1  function [ Sr ] = m_Sr(illuminators, receivers, target)
2  % Inputs
3  %   illuminators: M x 1 Illuminator object array
4  %   receivers: N x 1 Receiver object array
5  %   target: Target object
6  % Outputs
7  %   Sr: The intermediate matrix of the waveform derivates over receiver distances.
       (99)
8
9  % Initiate necessary indices
10 M = length(illuminators);
11 N = length(receivers);
12
13 % Preallocate a zero matrix of appropriate size
14 Sr = zeros(Constants.K*N,M*N);
15 % Loop over all illuminator-receiver pairs to generate elements
16 for m = 1:M
17     for n = 1:N
18         Sr(:,M*(n-1)+m) = m_Sri(illuminators(m),receivers(n),target,n,N);
19     end
20 end
```

```
1  function [ Sri ] = m_Sri(illuminator, receiver, target, n, N)
2  % Inputs
3  %   illuminator: Illuminator object
4  %   receiver: Receiver object
5  %   target: Target object
6  %   n: The Index of receiver
7  %   N: Number of receivers in the system
8  % Outputs
9  %   Sri: ith Column vector of Sr. (99)
10
11 % Initiate the parameters from object and known constants
12 c_r = receiver.Coordinates;
13 c_i = illuminator.Coordinates;
14 c_t = target.Coordinates;
15 E_m = illuminator.Transmit_energy;
16 T_s = Constants.T_s;
17 K = Constants.K;
18 P_0 = Constants.P_0;
19
20 % Preallocate a zero column vector of the appropriate size
21 p = zeros(K,1);
```

```
% Loop over time samples
for k = 1:K
    p1 = -sqrt((E_m*P_0)/(norm(c_r-c_t)^2*norm(c_i-c_t))) * exp(1i*2*pi*par_f(
        target,illuminator,receiver)*k*T_s);
    p2 = illuminator.Normalization_factor * d_non(k*T_s-par_tau(c_t,c_i,c_r),Wave(
        illuminator));
    p(k) = p1 * p2;
end

% Reshape the product to the final column matrix
e_n = full(sparse(n,1,1,N,1));
Sri = kron(e_n,transpose(p));
```

```
function [ Stau ] = m_Stau(illuminators, receivers, target)
% Inputs
%   illuminators: M x 1 Illuminator object array
%   receivers: N x 1 Receiver object array
%   target: Target object
% Outputs
%   Stau: The intermediate matrix of the waveform derivates over time delays.
    (85-86)

% Initiate necessary indices
M = length(illuminators);
N = length(receivers);

% Preallocate a zero matrix of appropriate size
Stau = zeros(Constants.K*N,M*N);
% Loop over all illuminator-receiver pairs to generate elements
for n = 1:N
    for m = 1:M
        Stau(:,M*(n-1)+m) = m_Staui(illuminators(m),receivers(n),target,n,N);
    end
end
```

```
function [ Staui ] = m_Staui(illuminator, receiver, target, n, N)
% Inputs
%   illuminator: Illuminator object
%   receiver: Receiver object
%   target: Target object
%   n: The Index of receiver
%   N: Number of receivers in the system
% Outputs
%   Staui: ith Column vector of Stau. (79)

% Initiate the parameters from object and known constants
c_r = receiver.Coordinates;
c_i = illuminator.Coordinates;
c_t = target.Coordinates;
E_m = illuminator.Transmit_energy;
T_s = Constants.T_s;
K = Constants.K;
P_0 = Constants.P_0;

% Preallocate a zero matrix of appropriate size
p = zeros(K,1);
% Loop over time samples
for k = 1:K
    p1 = sqrt((E_m*P_0)/(norm(c_r-c_t)^2*norm(c_i-c_t)^2)) * exp(1i*2*pi*par_f(
        target,illuminator,receiver)*k*T_s);
```

```
25      p2 = illuminator.Normalization_factor * d_tau(k*T_s-par_tau(c_t,c_i,c_r),Wave(
            illuminator));
26      p(k) = p1 * p2;
27  end
28
29  % Reshape the product to the final column matrix
30  e_n = full(sparse(n,1,1,N,1));
31  Staui = kron(e_n,transpose(p));
```

```
1   function [ St ] = m_St(illuminators, receivers, target)
2   % Inputs
3   %   illuminators: M x 1 Illuminator object array
4   %   receivers: N x 1 Receiver object array
5   %   target: Target object
6   % Outputs
7   %   St: The intermediate matrix of the waveform derivates over illuminator
        distances. (91)
8
9   % Initiate necessary indices
10  M = length(illuminators);
11  N = length(receivers);
12
13  % Preallocate a zero matrix of appropriate size
14  St = zeros(Constants.K*N,M*N);
15  % Loop over all illuminator-receiver pairs to generate elements
16  for n = 1:N
17      for m = 1:M
18          St(:,M*(n-1)+m) = m_Sti(illuminators(m),receivers(n),target,n,N);
19      end
20  end
```

```
1   function [ Sti ] = m_Sti(illuminator, receiver, target, n, N)
2   % Inputs
3   %   illuminator: Illuminator object
4   %   receiver: Receiver object
5   %   target: Target object
6   %   n: The Index of receiver
7   %   N: Number of receivers in the system
8   % Outputs
9   %   Sti: ith Column vector of St. (91)
10
11  % Initiate the parameters from object and known constants
12  c_r = receiver.Coordinates;
13  c_i = illuminator.Coordinates;
14  c_t = target.Coordinates;
15  E_m = illuminator.Transmit_energy;
16  T_s = Constants.T_s;
17  K = Constants.K;
18  P_0 = Constants.P_0;
19
20  % Preallocate a column vector of appropriate size
21  p = zeros(K,1);
22  % Loop over time samples
23  for k = 1:K
24      p1 = -sqrt((E_m*P_0)/(norm(c_r-c_t)*norm(c_i-c_t)^2)) * exp(1i*2*pi*par_f(
            target,illuminator,receiver)*k*T_s);
25      p2 = illuminator.Normalization_factor * d_non(k*T_s-par_tau(c_t,c_i,c_r),Wave(
            illuminator));
26      p(k) = p1 * p2;
27  end
```

```
28
29 % Reshape the product to the final column matrix
30 e_n = full(sparse(n,1,1,N,1));
31 Sti = kron(e_n,transpose(p));
```

```
1  function [ f ] = par_f( target,illu,receiver )
2  % Inputs
3  %   target: Target object
4  %   illu: Illuminator object
5  %   receiver: Receiver object
6  % Outputs
7  %   f: The Doppler shift of target relative illuminator-receiver pair in Hz [3.6]
8
9  v_target = target.Velocities;
10 c_target = target.Coordinates;
11 c_illu = illu.Coordinates;
12 c_receiver = receiver.Coordinates;
13
14 part1 = v_target(1)*(c_illu(1)-c_target(1)) + v_target(2)*(c_illu(2)-c_target(2)) +
       v_target(3)*(c_illu(3)-c_target(3));
15 part2 = v_target(1)*(c_receiver(1)-c_target(1)) + v_target(2)*(c_receiver(2)-
      c_target(2)) + v_target(3)*(c_receiver(3)-c_target(3));
16 f = part1/(Constants.lambda*par_d(c_target,c_illu)) + part2/(Constants.lambda*par_d
      (c_target,c_receiver));
```

```
1  function [ tau ] = par_tau( c_target,c_illu,c_receiver )
2  % Inputs
3  %   c_target: Target object or target coordinate row vector
4  %   c_illu: Illuminator object or illuminator coordinate row vector
5  %   c_receiver: Receiver object or receiver coordinate row vector
6  % Outputs
7  %   tau: The time delay of target relative illuminator-receiver pair in Hz [3.7]
8
9  % Input validation conditions
10 if ~isfloat(c_target)
11     c_target = c_target.Coordinates;
12 end
13 if ~isfloat(c_illu)
14     c_illu = c_illu.Coordinates;
15 end
16 if ~isfloat(c_receiver)
17     c_receiver = c_receiver.Coordinates;
18 end
19
20 tau = (par_d(c_target,c_illu) + par_d(c_target,c_receiver)) / Constants.c;
```

```
1  classdef Illuminator
2          % Class implementation for illuminator element.
3          % Not much more than a simple data container.
4          % Initialized with a 7 element row vector.
5
6      properties
7          Coordinates = [0 0 0];
8          Transmit_energy = 0;
9          Waveform_params = [0];
10         Normalization_factor = 1;
11     end
12
13     methods
14         function obj = Illuminator(c)
```

```
15            if nargin > 0
16                obj.Coordinates = c(1:3);
17                obj.Transmit_energy = c(4);
18                % NOTE at the moment specific to 3 parameters
19                            % This is waveform specific and need to be adjusted
20                            % to facilitate more sophicticated wave funtions.
21                obj.Waveform_params = c(5:7);
22                obj.Normalization_factor = f_normfactor(obj);
23            end
24        end
25
26
27        function [ factor ] = f_normfactor(illuminator)
28                        % Approximate wave normalisation.
29             avgover = 100;
30             p4 = 0;
31             for av = 1:avgover
32                 p3 = 0;
33                 for k = 1:Constants.K;
34                    kk = k*Constants.T_s + av*(1e-7);
35                    p2 = norm(d_non(kk,Wave(illuminator)));
36                    p3 = p3+(p2^2)*Constants.T_s;
37                 end
38                 p4 = p4 + 1/p3;
39             end
40             factor = p4/avgover;
41        end
42
43
44        function obj = set.Coordinates(obj,c)
45            assert(isfloat(c) && all(size(c)==[1 3]),'Coordinates must be 1 x 3
                  floating number array!')
46            obj.Coordinates = c;
47        end
48    end
49 end
```

```
1  classdef Receiver
2          % Class implementation for receiver element.
3          % Not much more than a simple data container.
4          % Initialized with a 3 element row vector
5          % containing the coordinates.
6
7     properties
8         % Dimensions [x, y, z]
9         Coordinates = [0 0 0];
10    end
11
12    methods
13        function obj = Receiver(c)
14            if nargin > 0
15                obj.Coordinates = c;
16            end
17        end
18
19        function obj = set.Coordinates(obj,c)
20            assert(isfloat(c) && all(size(c)==[1 3]),'Coordinates must be 1 x 3
                  floating number array!')
21            obj.Coordinates = c;
22        end
```

```
    end
end
```

```
classdef Target
        % Class implementation for target element.
        % Not much more than a simple data container.
        % Initialized with two row vectors: one for position, one for velocity.

    properties
        % Dimensions [x, y, z]
        Coordinates = [0 0 0];
        Velocities = [0 0 0];
    end

    methods

        function obj = Target(c, v)
            if nargin == 2
                obj.Velocities = v;
            end
            if nargin > 0
                obj.Coordinates = c;
            end
            % This is left unhold for array intialization
        end

        function obj = set.Coordinates(obj,c)
            assert(isfloat(c) && all(size(c)==[1 3]),'Coordinates must be 1 x 3
                floating number array!')
            obj.Coordinates = c;
        end

        function obj = set.Velocities(obj,v)
            assert(isfloat(v) && all(size(v)==[1 3]),'Velocities must be 1 x 3
                floating number array!')
            obj.Velocities = v;
        end
    end
end
```

```
classdef Wave
  % Class defining the waveform and necessary derivates.
        % This object dictates the waveform, thus this needs to
        % be rewritten for more complex wave functions.

     properties
        param = 0;
    end

    methods
        % Initialization is via passing an Illuminator object, and thus the
        % parameters used to descibe the waveform broadcasted.
        function obj = Wave(i)
            if nargin > 0
                obj.param = i.Waveform_params;
            end
        end

        % Non differential solution at an instant 'var'
        function v = d_non(var,W)
```

```
21              v = cos(2*pi*W.param(1)*var-(W.param(3)/W.param(2))*cos(2*pi*W.param(2)*
                    var));
22          end
23
24          % Simple derivation of the wave function in respect tau with the chain rule.
25          function v = d_tau(var,W)
26              v = sin(2*pi*W.param(1)*var - (W.param(3)/W.param(2))*cos(2*pi*W.param
                    (2)*var)) ...
27              * (2*pi*W.param(1) + (W.param(3)/W.param(2))*2*pi*W.param(2)*sin(2*pi*W.
                    param(2)*var));
28
29          end
30      end
31  end
```

APPENDIX C. GRAPHICAL RESULTS OF PARTIAL RECOMPUTING

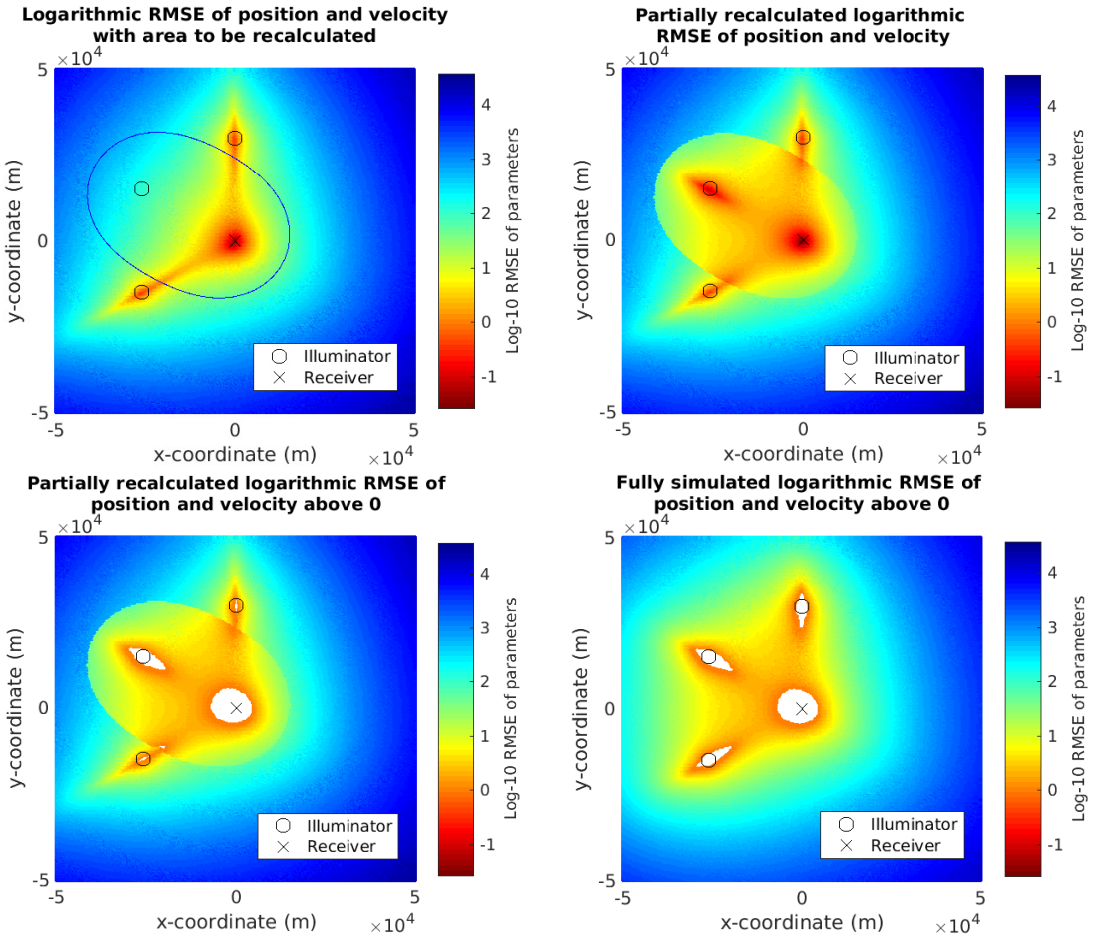

Figure 1 *Comparison between partly recalculated area and fully simulated area, when* $RMSE < 0$*, Cassini parameter is* 1.72*, and* β*-illuminator is added to* $\alpha\gamma$*-configuration*

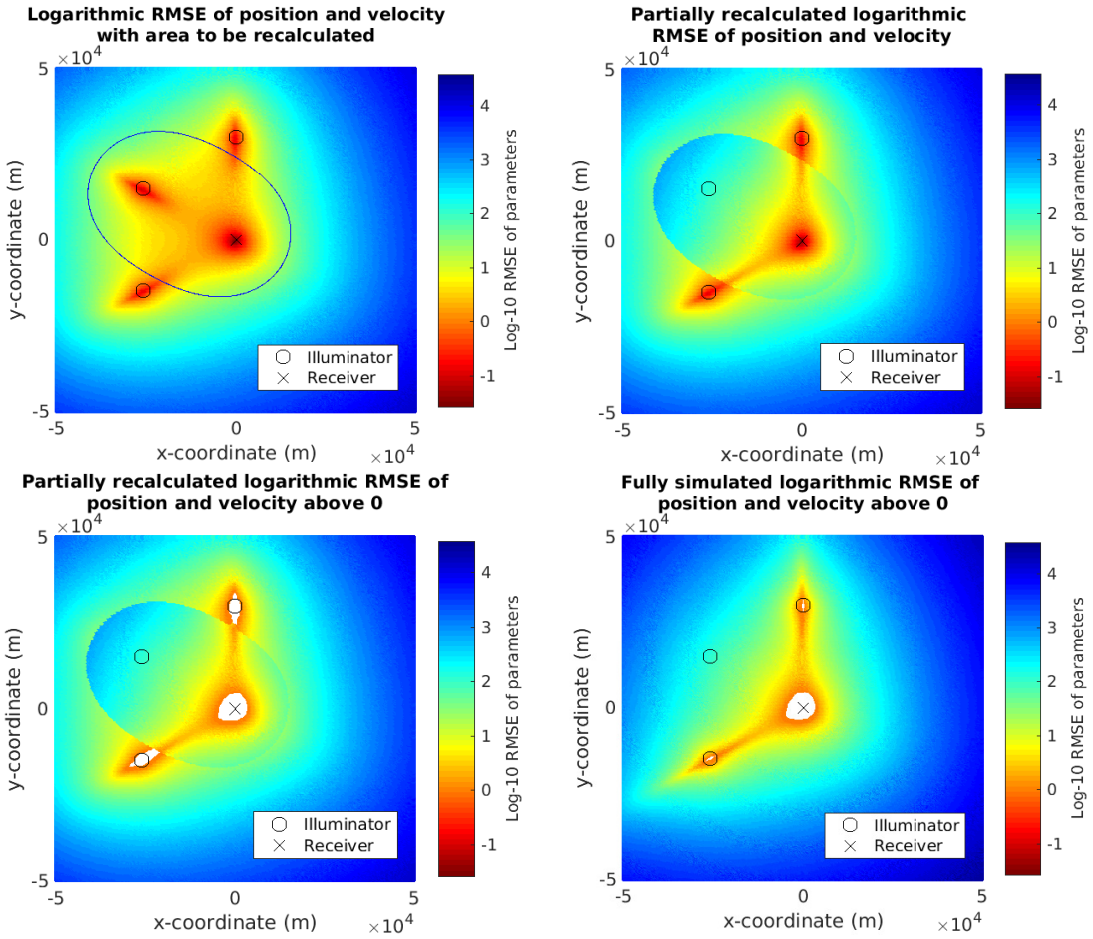

Figure 2 *Comparison between partly recalculated area and fully simulated area, when* $RMSE < 0$*, Cassini parameter is* 1.72*, and* β*-illuminator is removed from* $\alpha\beta\gamma$*-configuration*

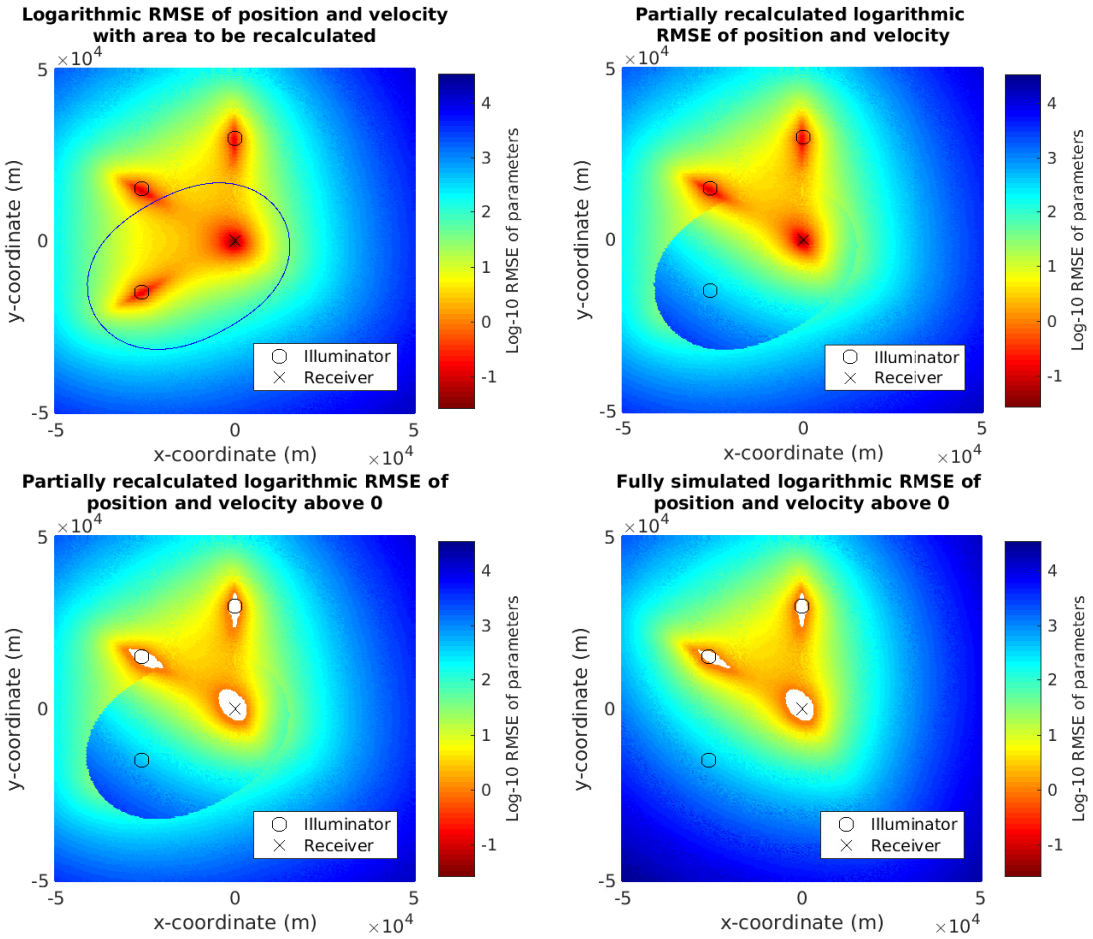

Figure 3 *Comparison between partly recalculated area and fully simulated area, when* $RMSE < 0$*, Cassini parameter is* 1.72*, and* γ*-illuminator is removed from* $\alpha\beta\gamma$*-configuration*

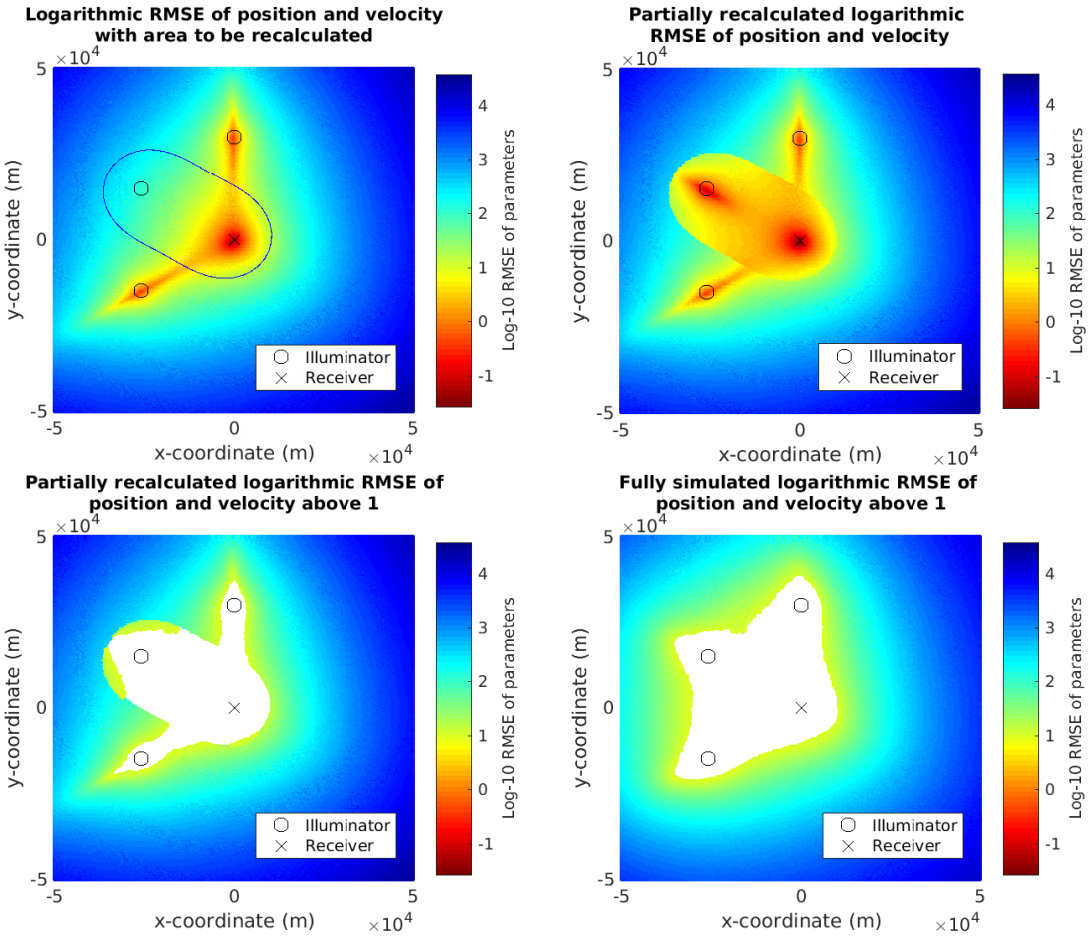

Figure 4 *Comparison between partly recalculated area and fully simulated area, when* $RMSE < 1$*, Cassini parameter is* 1.35*, and* β*-illuminator is added to* $\alpha\gamma$*-configuration*

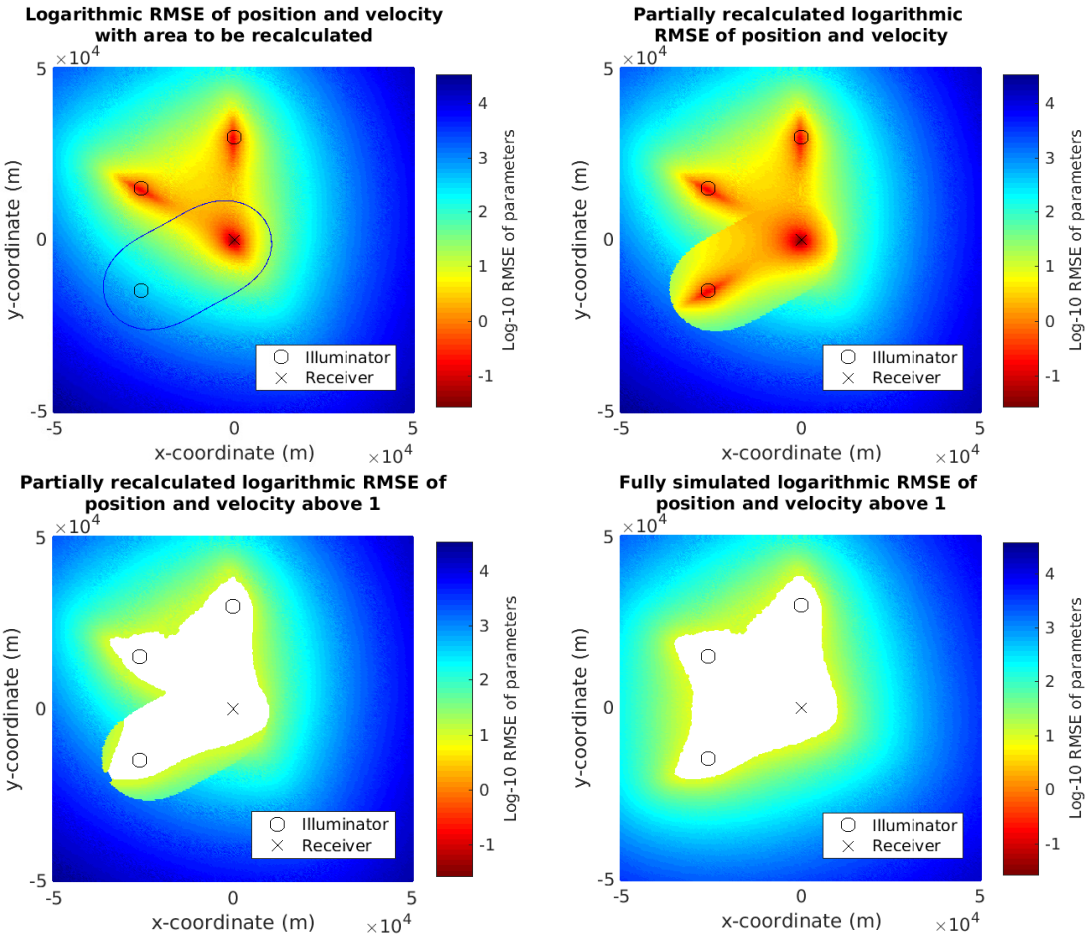

Figure 5 *Comparison between partly recalculated area and fully simulated area, when* $RMSE < 1$*, Cassini parameter is* 1.35*, and* γ*-illuminator is added to* $\alpha\beta$*-configuration*

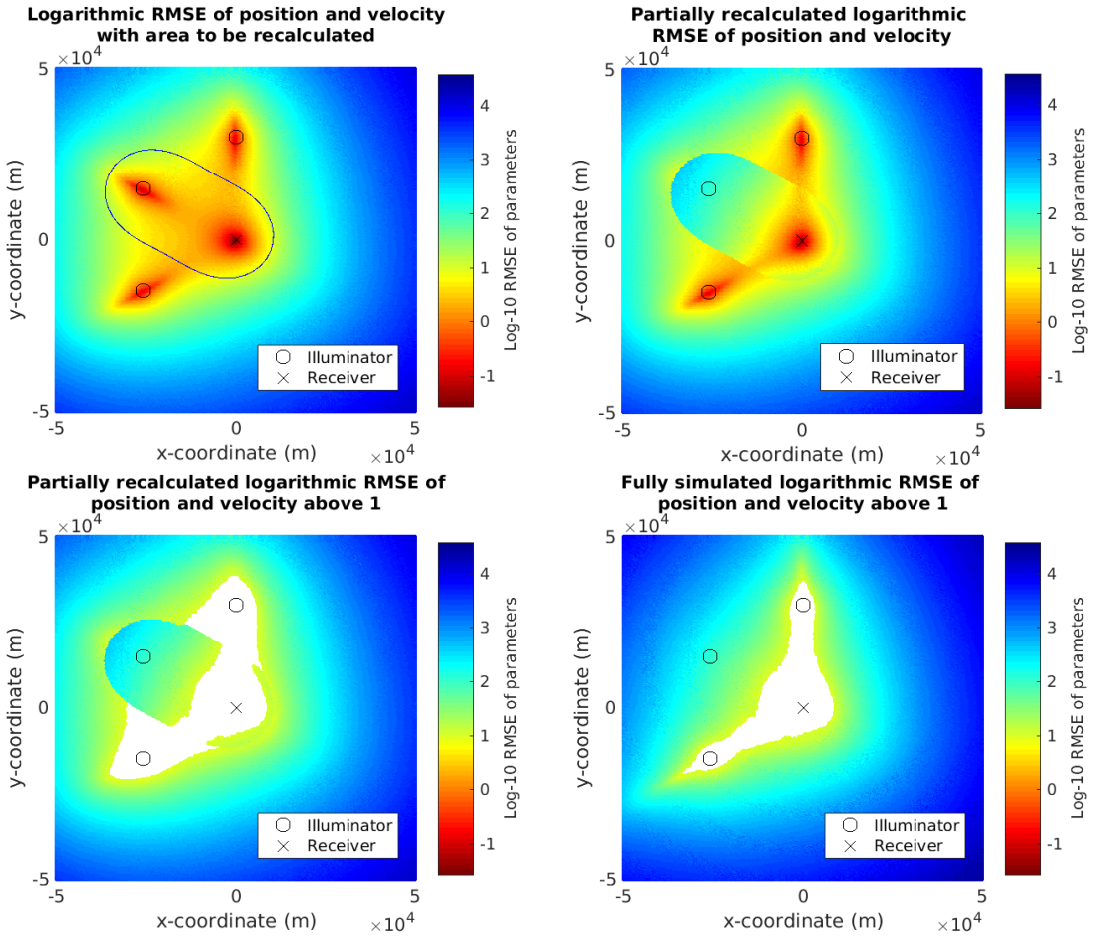

Figure 6 *Comparison between partly recalculated area and fully simulated area, when* $RMSE < 1$*, Cassini parameter is* 1.35*, and* β*-illuminator is removed from* $\alpha\beta\gamma$*-configuration*

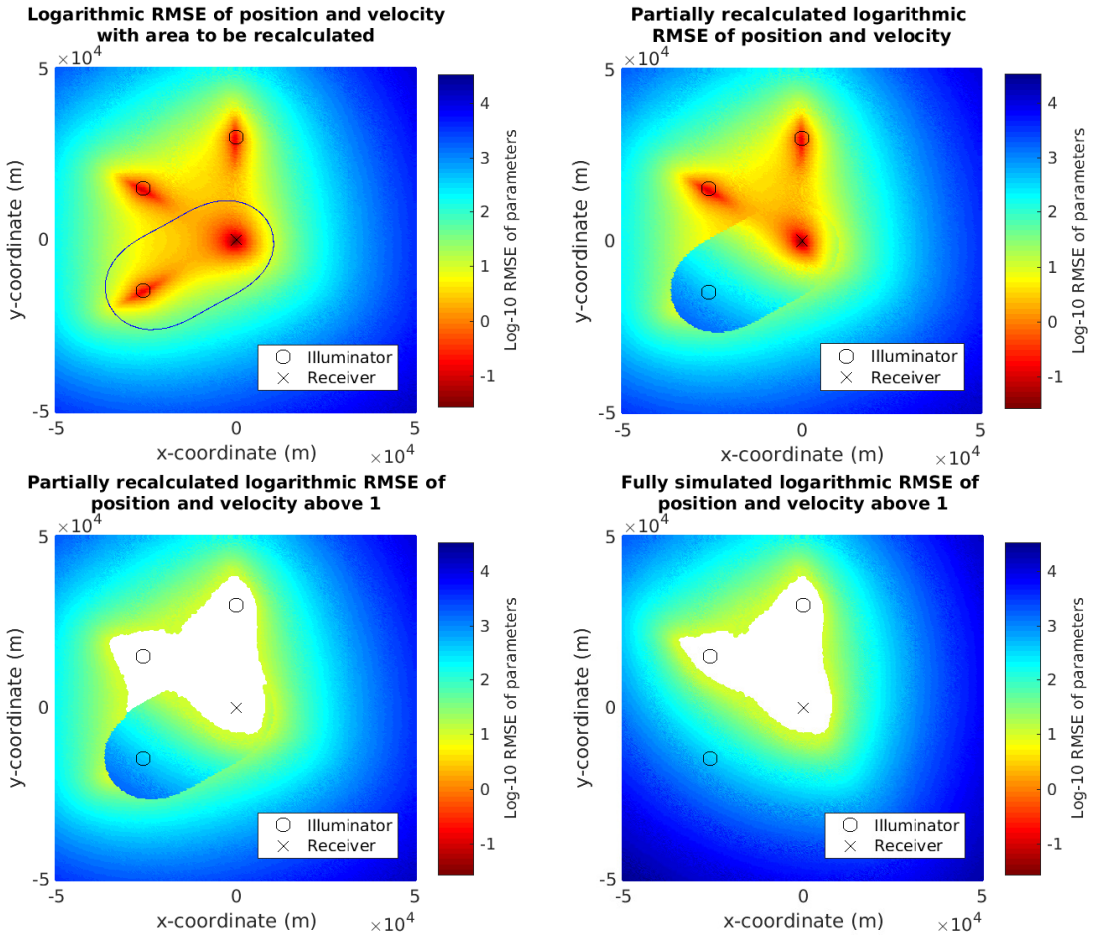

Figure 7 *Comparison between partly recalculated area and fully simulated area, when* $RMSE < 1$*, Cassini parameter is* 1.35*, and* γ*-illuminator is removed from* $\alpha\beta\gamma$*-configuration*

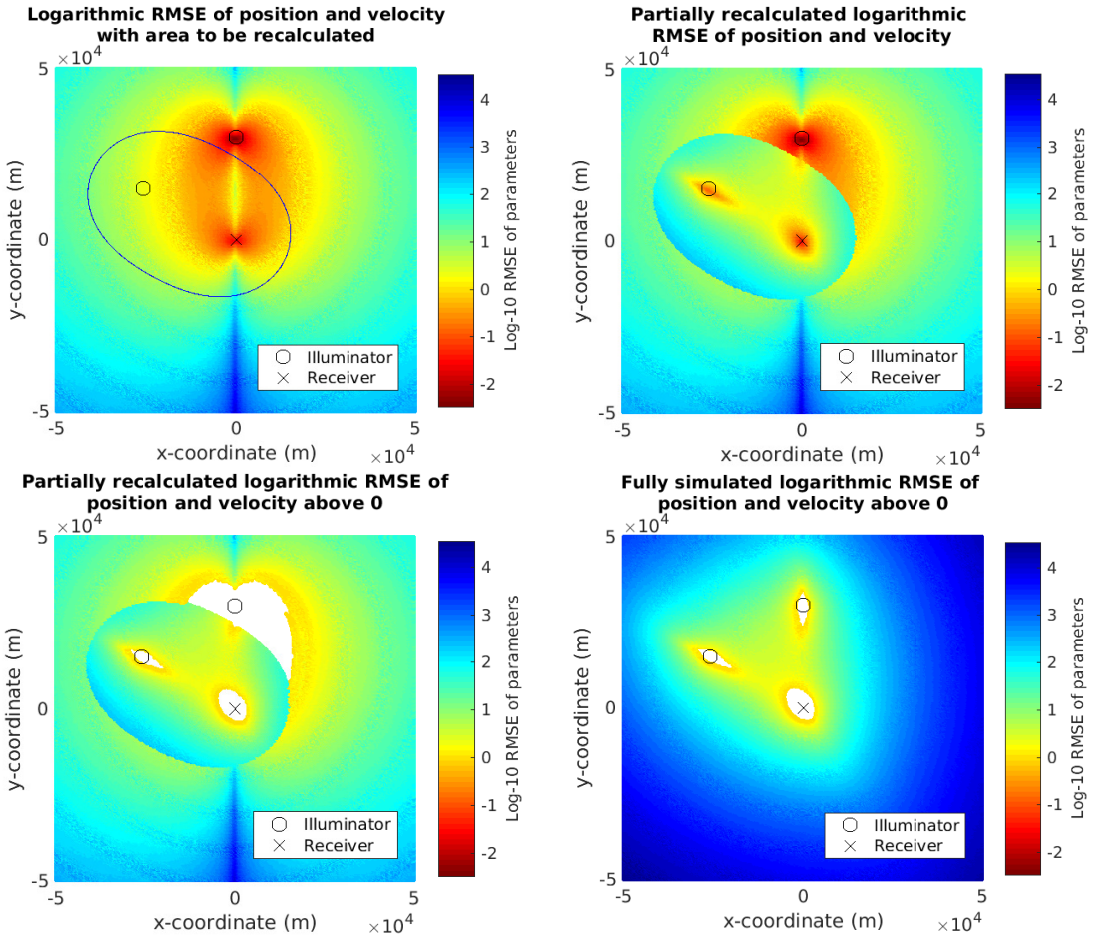

Figure 8 *Comparison between partly recalculated area and fully simulated area, when* $RMSE < 0$*, Cassini parameter is* 1.72*, and precomputed* β*-illuminator is added to single-illuminator* α*-configuration*

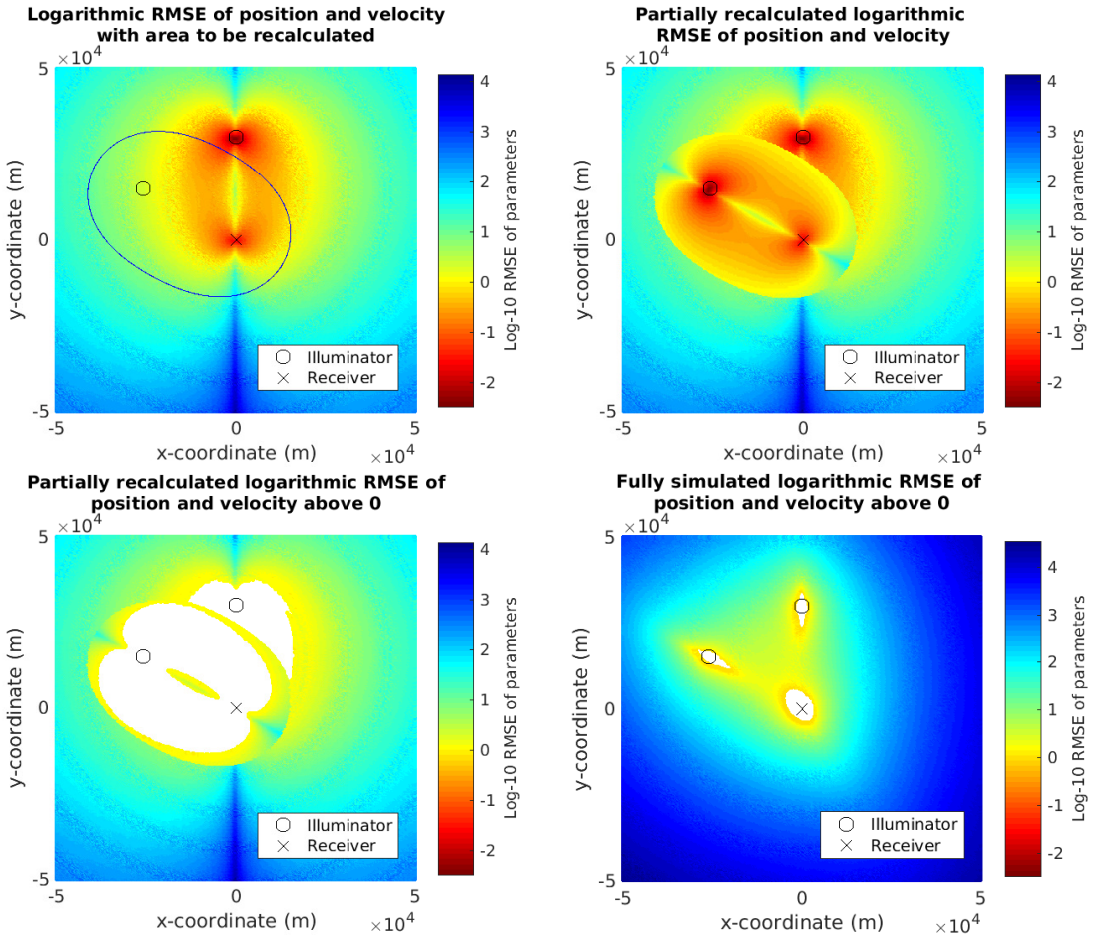

Figure 9 *Comparison between partly recalculated area and fully simulated area, when* $RMSE < 0$*, Cassini parameter is* 1.72*, and precomputed* β*-illuminator is added to single-illuminator* α*-configuration*